details of the new query are not disclosed, the new query (2) could theoretically be, *e.g.*, (a) a different portion of the same song, or (b) a better hummed version of the same portion of the same song.

288. If the second new query is viewed as a second separate search, each independent search would be exhaustive because (a) as I explained above, the initial search compares the query to all possible matches in the database—"all the songs" (Ghias, 5:66-6:2; Moulin Depo. 336:9-15) and (b) the restricted search also compares the query to all "possible matches" because the search compares the new query to all potential matches (illustrated by green dataset in the diagram above). The records that are not on the restricted list (*i.e.*, in the blue dataset but not the green dataset) are not "possible matches" for the restricted search. The first search excludes from the list of ranked songs those songs that are not possible matches such that the "restricted search list" comprises "all possible matches." Moulin Depo. 336:13-327:6; 335:13-336:12.

289. The only algorithm Ghias teaches for conducting a search is to compare a query statement against every record in the data set against which the algorithm is to be run—and is thus always an exhaustive search. Accordingly,

---

two stages because the query does not change (*i.e.*, the first view I addressed above).

whether the two-stage search is viewed as a single search or two separate searches, the searches compare the work to be identified with "all possible matches" and are therefore exhaustive searches.

290. The Board also noted that if Ghias disclosed an exhaustive search, Ghias would still disclose this element if Ghias also disclosed a non-exhaustive search:

> Additionally, given the "comprising" language used in the independent claims, we are not persuaded that the claimed methods could not cover processes with both exhaustive and non-exhaustive searching, as long as the latter provides identification.

Decision ('988) at 12. Because, as I described above, Ghias does not disclose any non-exhaustive search, Ghias does not anticipate.

2. search identifying a neighbor (claim element 15(b)).

291. In instituting Ground 1, the Board did not specifically find that Ghias disclosed a search identifying a neighbor. Decision ('988) at 12.

292. As I explained in detail above, a search identifying a neighbor means a search identifying "a close, but no necessarily exact or closest, match." Section V(C) ¶¶X; Decision ('988) at 12.

293. As I explained above in detail, Ghias does not disclose a search that identifies a neighbor because the searches disclosed in Ghias always identify an

exact or the closest match. Ghias teaches a search that generates three possible outputs:

(1) an exact match (Ghias 2:53-59 ("exact matching melody"));

(2) a "ranked list of approximately matching melodies" (Ghias, 2:50-59; Ghias, 6:60-63 ("a list of songs ranked by how well they matched the query"); Moulin Depo. 118:9-22); or

(3) "the single most approximate matching melody" (Ghias, 2:50-59).

As I demonstrated above, for each output, the Ghias search necessarily identifies an exact or closest match. Moulin Depo. 352:22-353:2. Accordingly, Ghias does not disclose a search "identifying a neighbor."

294. The Petition and corresponding Declaration fail to demonstrate that Ghias discloses a search "identifying a neighbor."

295. Petition: As support for the claimed "identifying a neighbor," I note that the Petition relies on the following:

> Ghias further discloses that this search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." Ex. 1010 at 2:50-59, 6:60-63.

Pet. ('988) 10.

296. Petition Charts: The charts in the Petition the same assertions and passages from Ghias:

Petitioner's chart for claim 15, element [c] incorporates the chart for claim 1, element [c]:

b) electronically determining an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor,	Petitioner incorporates the above discussion of Ghias regarding Claim 1c

Pet. ('988) at 14. The chart for claim 1, element [c], in turn, provides:

c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;	Ghias receives and outputs at the computer, which is a portable client device (Ex. 1004 at ¶ 73), a list of identifications of electronic works. 2:50-52, 6:60-63, 7:4-5, 8:26-28, 8:61-63. Such identifications are determined by "searching the melody database 14" to locate a matching melody. 2:50-59, 6:60-63, 7:4-5, Abstract, 8:26-28, 8:61-63. This search may employ a non-exhaustive "approximate pattern matching algorithm" or another algorithm that operates faster than a brute force search. 6:7-11, 6:23-35. This non-exhaustive search identifies a neighbor, i.e., "a ranked list of approximately matching melodies." 2:50-59, 6:60-63.

Pet. ('988) at 12.

297. Declaration: Petitioner's Declaration relies on the same assertion and passages from Ghias:

> 69. Second, it is my opinion that Ghias discloses "electronically determining an identification of the electronic work based on the extracted features" by performing "a non-exhaustive search identifying a neighbor." Ex. 1001 at Claims 1, 15. In particular, Ghias discloses using a "query engine 24" which "searches the melody database 14" to locate a matching melody. Ex. 1010 at 2:50-59, Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences between successive notes which at least approximately matches the sequence of digitized representations of relative pitch differences between successive notes of the melody."). Ghias further

Moulin Decl. ('988) ¶¶69-70.

298. Declaration Charts: Finally, the charts in the declaration also rely on the same two assertions and the same two passages from Ghias:

c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;	Ghias discloses receiving and outputting at the computer, which is a portable client device, a list of identifications of electronic works. 2:50-52, 6:60-63, 7:4-5, 8:26-28, 8:61-63. Ghias further discloses that such identifications are determined by "searching the melody database 14" to locate a matching melody. 2:50-59, 6:60-63, 7:4-5, Abstract, 8:26-28, 8:61-63. Ghias further discloses that this search may employ a non-exhaustive "approximate pattern matching algorithm" or another algorithm that operates faster than a brute force search. 6:7-11, 6:23-35. Ghias further discloses that this non-exhaustive search identifies a neighbor by determining "a ranked list of approximately matching melodies." 2:50-59, 6:60-63.

Moulin Decl. ('988) ¶75.

299. Neither the assertions nor the passages from Ghias disclose the claimed non-exhaustive search because, as described in detail above, both the ranked list and single most approximate matching melody outputs always identify the closest match.

3. determining an action based on the identification (claim element 15(c)).

300. The instituted claims are not anticipated by Ghias because the Petition fails to demonstrate that Ghias discloses "determining an action based on the identification of the electronic work" and "performing the action."

301. Claim 15—the only instituted independent claim—claims a method comprising four steps. Steps (b) through (d) are:

> b) electronically determining an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;
> c) electronically determining an action based on the identification of the electronic work; and
> d) electronically performing the action.

As reflected in the claim, "determining an action" in step (c) and "performing the action" in step (d) must be based on the "identification of the electronic work" from step (b).

302. It is my understanding that the third and fourth steps—(c) determining an action based on the identification, and (d) performing the action—must have meaning beyond that encompassed by step (b) "determining an identification" because all limitations in a claim must be considered meaningful.

303. The Petition asserts that Ghias discloses step (c)—"determining an action based on the identification"—in two ways:

(1) "using the result of a search to determine the potential matches;" and

(2) "allowing the user to perform a 'new query on a restricted search list consisting of songs just retrieved."

Pet. ('988) at 10; 12; Moulin Decl. ('988) ¶¶71, 75. Neither discloses "determining an action based on the identification" for at least two reasons.

304. First, according to Petitioner, both (a) "determin[ing] the potential matches," and (b) performing a search on a restricted list constitute the second step (b)—"determining an identification of the electronic work." In claim 15, determining the action must be "based on the identification." Accordingly, the system must first identify the electronic work, and then, based on the identity of the work, "determine an action."

(1) "[D]etermine the potential matches" is part of "determining an identification of the electronic work" and therefore cannot be an action based on the identification of the electronic work.

(2) Similarly, performing a search on a restricted list of potential matches is identified as part of "identification of the electronic work."

See Decision ('988) at 12 ("Ghias provides that '[t]he number of matches that the database 14 should retrieve depends upon the error-tolerance used during the key-search.' Ex. 1010, 6:63–65 (emphasis added). Ghias further provides that 'the user can perform a new query on a restricted search list consisting of songs just retrieved. This allows the user to identify sets of songs that contain similar melodies.' Id. at 7:5–8 (emphasis added).") If the search on the restricted list is part of the search identifying the match, it cannot also be the action based on that search. Accordingly, the Petition fails to identify steps (c) and (d) in Ghias.

305. Second, Ghias does not determine an action "based on the identification of the electronic work." One skilled in the art would understand that, to be "based on" the identification, the action must depend upon the identification. In Ghias, the actions identified by Petitioner are performed independent of the identification.

(1) "[D]etermine the potential matches" is part of the process of identifying the work and therefore is not based on the identification of the electronic work.

(2) Similarly, performing a search on a restricted list of potential matches is not based on the identification of the work.

Rather, whether a new query is performed on a restricted search list is solely a consequence of the number of potential matches, not the identity of the matches. Thus, this action performed by Ghias is the same regardless of the identity of the electronic work.

B. '988 Ground 2: The instituted claims of the '998 patent are not obvious over Ghias.

306. Ground 2 relies exclusively on Ghias and is directed to only dependent claims 22, 24-26, and 52, which depend (indirectly) on independent claim 15. Pet. ('988) at 54-57; Decision ('988) at 22.

307. As I demonstrated above, Ghias does not disclose elements from the independent claim upon which Ground 2 is based (claim 15) including:

- "non-exhaustive search identifying a neighbor" (claim element 15(b));
- "electronically determining an action based on the identification of the electronic work" (claim element 15(c)); and
- "electronically performing the action" (claim element 15(d)).

308. In Ground 2, Petitioner (and the Board) do not assert that these missing elements are obvious in light of Ghias but rather assert that these missing elements are expressly disclosed in Ghias. *See e.g.*, Pet. ('988) at 55 ("For the reasons expressed in Ground 1 [anticipated by Ghias], Ghias discloses all elements of claims 1 and 15."). Accordingly, Ground 2 fails because the elements from the

independent claims addressed above are missing from Ghias and the Petition does not provide any basis for correcting these deficiencies.

C. '988 Ground 3: The instituted claims of the '998 patent are not anticipated by Iwamura.

309. The single independent claim of the '988 patent instituted for trial includes the phrase "non-exhaustive search identifying a neighbor." Claim 15. Iwamura does not anticipate the instituted claims because Iwamura does not disclose: (1) a "non-exhaustive" search; and (2) "identifying a neighbor." I address each deficiency in turn.

1. non-exhaustive search (claim 15(b)).

310. Iwamura does not disclose the claimed "non-exhaustive" search.

311. As I explained above, a non-exhaustive search is "a search that locates a match without a comparison of all possible matches." Section V(B); Decision ('988) at 7.

312. Iwamura does not disclose "a search that locates a match without a comparison of all possible matches." As I explained above in detail, Iwamura discloses a searching algorithm that is designed to be more efficient than alternatives by lining up peak notes from the music work to be identified with the peak notes in each record in the music database, when comparing the work to each

record. Iwamura, 6:59-60; 12:1-2. Instead of comparing the work to be identified with a record in the database by (a) performing a first comparison of the notes in the work and the record, and then (b) shifting the comparison between the work and the record "note by note" to see if there is a match, the shifting can be done peak-note-to-peak-note, thereby reducing the number of comparison made between the work and a specific record, and thus making the comparison more efficient.

> "Peak notes are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search which shifts the entered melody, note by note."

Iwamura, 9:9-11.

313. As I explained above in detail:

- each melody in the melody database is compared using this peak note approach and "[t]he reference melody that gives the least difference is returned as a search result" (Iwamura, 7:54-55);
- Petitioner's Declarant confirmed that "for all the Iwamura searches…[i]t's understood that you search through every musical work in the database"—*i.e.*, all potential matches (Moulin Depo. 269:19-270:2; 223:2-8; 247:18-20; 271:19-21; 207:18-23); and
- Petitioner's Declarant also confirmed that "all the notes" are compared (Moulin Depo. 280:6-13; 277:6-21).

As a result, Petitioner's Declarant confirmed that, based on the proper construction of non-exhaustive search (adopted by the Board) and the understanding of one of ordinary skill in the art, Iwamura does not disclose a non-exhaustive search. Moulin Depo. 233:24-234:14; 225:16-226:7; 217:1-18.

314. The Petition and corresponding Declaration fail to demonstrate that Iwamura discloses a "non-exhaustive" search. I note that Petitioner and its Declarant identify three features of the Iwamura search as teaching non-exhaustive searching:

(a) peak notes: a search that uses peak notes, which are approximately 20% of the total number of notes in a typical melody;"

(b) limit function: a search in which a specific comparison of the work to be identified to a specific record in the database "can be stopped," when the specific computation of the total absolute difference between the work to be identified and the specific record exceeds a certain limit;

(c) unsearched portions: a search that skips "portions that should not be searched," such as "repeated patterns" and "unimportant melodies."

Pet. ('988) at 47-48.

Petitioner identifies these three features (labeled ❶, ❷, and ❸) as disclosing the non-exhaustive search in its Petition, Declaration, and corresponding charts:

315. Petition:

> Iwamura further teaches how this search can be non-exhaustive. For example, Iwamura teaches a non-exhaustive search using "peak notes" (Ex. 1012 at 6:31-7:55) which "are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search." *Id.* at 9:8-11. In another example of non-exhaustive searching, Iwamura teaches that the search can be accelerated by stopping the search when computations "exceed[] a certain limit." Ex. 1012 at 7:56-57. In yet another example of non-exhaustive searching, Iwamura discloses skipping "portions that should not be searched." *Id.* at 12:6-7. These skipped portions include "repeated patterns" (*id.* at 9:36-44), and "unimportant portion[s]" of the melody (*id.* at 9:44-45).

Pet. ('988) at 47-48.

316. Petition chart: Claim 15 (the instituted claim, cross-references claim 1, element (c)):

b) electronically determining an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;	Petitioner incorporates the above discussion of Iwamura regarding Claim 1c.

Pet. ('988) at 52. Claim 1, element (c):

e) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor.	Different "search algorithms may be applied to perform melody searches" (10:1-3), including non-exhaustive searches, such as "peak notes" ❶ (6:31-7:55) which "are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search" (9:8-10). Other disclosed non-exhaustive searches can decrease search time by stopping the search when computations ❷ "exceed[] a certain limit" (7:56-57) and can be configured to skip "portions that should not be searched," (12:6-9), ❸ such as "repeated patterns," (9:36-44), and "unimportant portion[s]" of the melody. (9:44-45)

Pet. ('988) at 50.

317. <u>Declaration</u>:

> 141. It is my opinion that Iwamura further teaches how this search can be non-exhaustive. For example, Iwamura teaches a non-exhaustive search that uses "peak notes." Ex. 1012 at 6:31-7:55. "Peak notes are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search . . . ." *Id.* at 9:8-11.
>
> 142. It is my opinion that Iwamura's disclosure that the search can be accelerated by stopping the search when computations "exceed[] a certain limit" is another example of non-exhaustive searching. Ex. 1012 at 7:56-57.
>
> 143. It is my opinion that Iwamura's disclosure of skipping "portions that should not be searched" (Ex. 1012 at 12:6-7) wherein these skipped portions include "repeated patterns" (*id.* at 9:36-44) and "unimportant portion[s]" of the melody (*id.* at 9:44-45) constitutes another example of non-exhaustive searching.

Moulin Decl. ('988) ¶¶141-143.

318. Declaration chart: Claim 15 (the instituted claim, cross-references claim 1, element (c):

b) electronically determining an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor.	I incorporate my above discussion of Iwamura regarding Claim 1c.

Moulin Decl. ('988) ¶147.

Claim 1, element (c):

c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor.	Iwamura discloses that "[t]he web server returns the search result to the client." 5:1. Iwamura discloses the use of a "search engine" to "find the closest melody from the database." 9:23. Iwamura discloses using "a peak or differential matching algorithm." 12:1-2. Iwamura discloses that different "search algorithms may be applied to perform melody searches." 10:1-3. Iwamura discloses non-exhaustive searches identifying a neighbor, such as "peak notes" ❶ (6:31-7:55) which "are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search ..." (9:8-10). Iwamura further discloses non-exhaustive search identifying a neighbor that can be accelerated by stopping the search when computations "exceed[] a certain limit" ❷ (7:56-57) and can be configured to skip "portions that should not be searched." ❸ (12:6-9), such as "repeated patterns," (9:36-44), and "unimportant portion[s]" of the melody. (9:44-45)

Moulin Decl. ('988) ¶147.

319. As I explained above in detail, none of these three Iwamura search features (peak notes, limit function, or unsearched portions) relied on by Petitioner and its Declarant for the non-exhaustive search discloses the claimed non-exhaustive search.

320. Board's concerns: I now address the Board's specific concerns (identified in its Decision in the '988 IPR) with respect to whether Iwamura discloses the claimed non-exhaustive search. In instituting Ground 3, the Board preliminary determined that one feature of Iwamura identified by Petitioner—the "computational limits" feature—discloses a non-exhaustive search because if the

computational limit is reached, the entire search is stopped, even if all of the records have not been searched:

> Petitioner identifies Iwamura's computational limit as an example of non-exhaustive searching, in that not all records in the remote music database necessarily are searched. Pet. 48. Patent Owner argues that Iwamura's description of stopping a search when computations exceed a certain limit is not a non-exhaustive search because "it does not state or suggest that all records in the music library are not used in the comparison." Prelim. Resp. 27. We do not agree. If, in Iwamura, the computational limit is reached, the search is stopped, even if not all of the records have been searched. Per our construction of "non-exhaustive search," i.e., "a search that locates a match without a comparison of all possible matches," we are persuaded on this record that the process of Iwamura, with the computational limit, would prevent all of the records of the remote music database from being searched, but ultimately would provide a match using an input fault tolerance process to find the closest melody. *See* Ex. 1012, 7:56–57, 9:20–34.

Decision ('988) at 15.

321. As I explained above in detail, in making this preliminary finding, the Board apparently confused:

(a) stopping an individual computation of the absolute difference between the notes in the work to be identified with a specific record in the database and then shifting the peaks to do another comparison with that record, or moving on to the next record, with

(b) stopping the search process altogether.

And as I explained above in detail, there are two reasons why the Board's preliminary interpretation of Iwamura is wrong.

322. Reason 1: Iwamura never states (or even suggests or implies) that when a given computation (the absolute difference between the compared notes) based on comparing a work to be identified with a specific record in the database exceeds a certain limit (demonstrating that the particular alignment of work to be identified with the specific record being searched is not a match), the search stops.

323. Reason 2: The alternative (which is not identified in Iwamura)—that the entire search stops when one peak search comparison between the work to be identified and one record in the database reaches a certain limit—make no sense.

324. The Board also noted that if Iwamura disclosed a non-exhaustive search, Iwamura would still disclose this claimed element even if Iwamura also disclosed an exhaustive search:

> We note that claim 15 utilizes "comprising" language, such that the claimed method does not exclude additional, unrecited steps. *See Mars Inc. v. H.J. Heinz Co.*, 377 F.3d 1369, 1376 (Fed. Cir. 2004). Thus, the scope of independent claim 15 can include an exhaustive search, as long as it performs a non-exhaustive search as well. Thus, even if Patent Owner is correct and a particular search in Iwamura is exhaustive, that does not end the inquiry.

Decision ('988) at 14-15. Because, as described above, Iwamura does not disclose any non-exhaustive search, Iwamura does not disclose this element.

2. identifying a neighbor (claim 15(b)).

325. The Board did not previously address whether Iwamura discloses the claimed "identifying a neighbor." Decision ('988) 14-16. As I demonstrate below, the Petition and corresponding fails to establish that Iwamura discloses the claimed "identifying a neighbor."

326. As I explained above in detail, "identifying a neighbor" is a search that identifies "a close, but not necessarily exact or closest, match." Decision ('988) at 8.

327. Also, as explained above in detail, Iwamura does not disclose "identifying a neighbor" because the disclosed search always identifies an exact or the closest match. Petitioner asserts that Iwamura identifies a neighbor because "In Iwamura, once a melody has been extracted from the input, the 'search engine will find the closest melody from the database." Pet. ('988) 47 (*citing* Iwamura 9:24-25); Moulin Decl. ('988) ¶140. These statements do not establish a "neighbor search." Instead, they confirm that Iwamura always identifies the closest match—necessarily the closest match—rather than a match that is not necessarily the closest match, as required by the claimed "identifying a neighbor."

VIII. '179 patent.

328. The Board instituted the '179 IPR on two grounds:

- Ground 1: Claims 1–3, 6, 8–14, 19, 21–26, 30, 31, and 34–37 as unpatentable under 35 U.S.C. § 102(e) as anticipated by Conwell; and
- Ground 2: Claims 1–3, 8, 10–14, 18, 19, 21–27, 29, 31, and 34–37 as unpatentable under 35 U.S.C. § 103 as obvious over Ghias and Philyaw.

Decision ('179) at 15. I address each Ground in turn.

A. '179 Ground 1: The instituted claims of the '179 patent are not anticipated by Conwell.

329. I understand that to anticipate a claim, all elements of the claim need to be disclosed in a single prior art reference—in this case, Conwell. Each independent claim of the '179 patent includes a limitation "comparing [the extracted features of the work to be identified with extracted features of the reference works] using a non-exhaustive neighbor search:"

> (c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;

'179 claim 1, element [c];

> (c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation using a non-exhaustive neighbor search;

'179 claim 13, element [c];

> (c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation of the first electronic work using a non-exhaustive neighbor search;

'179 claim 25, element [c].

330. Ground 1 fails because Conwell does not disclose the claimed "comparing [the extracted features] using non-exhaustive neighbor search." The search disclosed in Conwell is neither (1) a "neighbor search," nor (2) a "non-exhaustive … search." I address each deficiency in turn.

1. neighbor search (claims 1, 13, 25).

331. The search disclosed in Conwell does not "compar[e] [the extracted features] using a …neighbor search."

332. As I explained above (Section V(C)), a "neighbor search" is a search "identifying a close, but not necessarily exact or closest, match." Section V(C); Decision ('179) at 8.

333. Each independent claim of the '179 patent requires comparing [1] the extracted features of the reference works to [2] the extracted features of the work to be identified, "using… a neighbor search."

Claim 1: "comparing"

> [1] "the extracted features of the first electronic work" with
>
> [2] the "first electronic data related to identification of one or more reference electronic works"

"using … a neighbor search;"

Claim 13: "comparing"

> [1] "the first electronic data" with
>
> [2] "second digitally created compact electronic representation of a first electronic work"

"using … a neighbor search";

Claim 25: "comparing"

> [1] "the first electronic data" ("comprising a first digitally created compact electronic representation comprising an extracted feature vector of one or more reference electronic work") with
>
> [2] "the second digitally created compact representation of the first electronic work"

"using a … neighbor search."[33]

334. Using claim 1 as an example:

> (c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;
>
> '179, claim 1

the claim requires "comparing… using a … neighbor search" (highlighted in yellow)

---

[33] In Petitioner's analysis, the "extracted features" ('179, claim 1) and "compact electronic representations" ('179, claims 13 and 25) are "synonymous"—both constitute the hashed identifiers of the extracted features of the underlying reference works in the database of reference works and works to be identified. Moulin Decl. ¶85 ("'compact electronic representation'—or, synonymously, 'extracted features' or a 'feature vector'"—of an unknown work.") Moulin Depo. 182:5-10 ("Q. You write, the term compact electronic representation is synonymous with extracted features… A. So in this context, it is accurate.") For simplicity, this Response refers to both the "extracted features" ('179, claim 1) and "compact electronic representations" ('179, claims 13 and 25) as the "extracted features."

[1] "the extracted features of the first electronic work" (highlighted in green); with

[2] "the first electronic data in the database" (highlighted in orange).

335. The claimed "comparing" does not compare the [1] work to be identified with [2] a record or records in the database. Using claim 1 as an example, the claimed "comparing… using [a] neighbor search" does not compare:

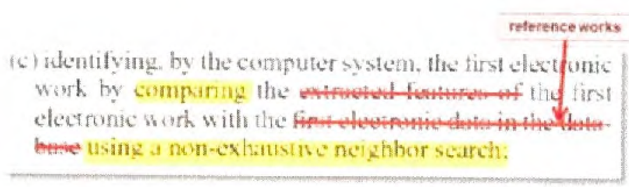
[1] "one or more reference electronic works" (*i.e.*, the references in the database); with

[2] "a first electronic work" (the work to be identified).

Such "comparing" is not reflected in the actual claim language but instead would require redrafting the '179 claims. For example, for claim 1:

- "the extracted features of the first electronic work" would need to be replaced with the "first electronic work;" and
- "first electronic data in the database" would need to be replaced with the underlying "reference electronic works,"

as reflected in the following redrafted language from claim 1:


reference works

(c) identifying, by the computer system, the first electronic work by comparing the ~~extracted features of~~ the first electronic work with the ~~first electronic data in the database~~ using a non-exhaustive neighbor search;

'179, claim 1

The redrafted claim is not the claimed invention. Instead, the claimed "comparing" requires comparing the extracted features—[1] the extracted features of the work to be identified with [2] the extracted features of the reference works—using a neighbor search.

336. Moreover, the claims do not claim a process that simply results in or has the effect of identifying a neighbor of the work to be identified from the reference works using any possible comparison or method. Rather, the claimed process requires comparing [1] the extracted features of the work to be identified, and [2] the extracted feature of the reference works "using [a] neighbor search." One skilled in the art at the time of the invention, would understand that the neighbor search would require that the extracted features be neighbors of the first electronic data, and not merely that the first electronic work be a neighbor of the identified referenced work.

337. The extracted features from Conwell identified in the Petition and Declaration (and relied on by the Board) that are compared are the hashes of the extracted features (the "identifiers") from the reference works to be identified and the records in the database. The Petitioner and Declarant (and the Board in instituting Ground 1) relied on the hashes of the extracted features of the work to be identified and the records in the database (reference works) in Conwell (the "identifiers") as the extracted features that are compared to establish the neighbor

search. *See* Pet. ('179) at 23 (the hashed "identifier extracted from a reference electronic work" are compared to the hashed "extract identifiers from content"); Moulin Decl. ('179) ¶85 ("Conwell further teaches that a hash algorithm can be selected so that 'similar, but non-identical, inputs map on the same hash outputs.'" (quoting Conwell, 4:64-5:3); Conwell, 1:65-67 ("some or all of the content data is processed by a hashing algorithm to yield a 128 bit identifier corresponding to that content.")

Q. To satisfy that limitation, does Conwell -- what does Conwell disclose that satisfies that limitation?
A. So the outputs -- this is line 66-67 in Conwell, Column 1. So the -- some of the -- "Some or all of the content data is processed by hashing algorithm to yield a 128-bit identifier corresponding to that content."
So you start from the content data, and you map that to a 128-bit identifier, which is a compact electronic representation of the content.

Moulin Depo. 180:7-17.

Is it your testimony that a 128-bit hash identifier constitutes a digitally created compact electronic representation?
A. Yes.

Moulin Depo. 180:22-25.[34]

338. The comparison relied on by the Petitioner and Declarant as disclosing the claimed "neighbor search" is illustrated in the following diagram:

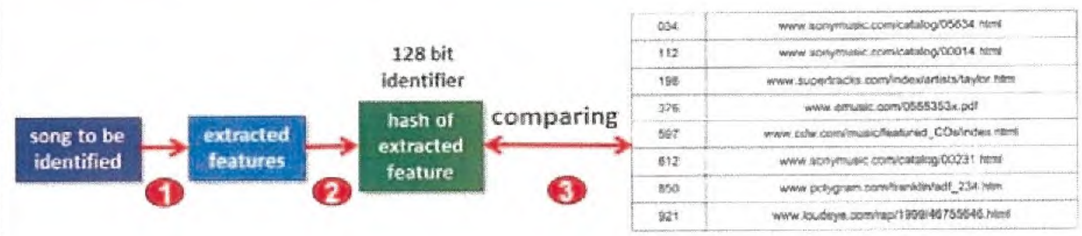


FIG. 3

[34] Conwell teaches a second alternative approach that "uses the bits of an audio work as an identifier and does not use a hash." Moulin Depo. 172:12-18. Conwell, 1:60-65 ("One way to derive an identifier is to employ selected bits of the content, itself, as the identifier.") Petitioner does not rely on this approach as disclosing the claimed non-exhaustive search because it does not teach searching—"[i]t does not talk at all about searching" much less the claimed neighbor search. Moulin Depo. 172:12-18. Moulin Depo. 184:19-23. The Declarant confirmed that using the bits as identifies (as opposed to the hashes) does not result in a neighbor search. Moulin Depo. 170:25-171:11 ("If we're to use only the bits as identifiers … [t]hen there's no neighbor search.")

First, features are extracted from the work—the song to be identified. Second, the extracted features are hashed to generate a 128 bit identifier. Third, the hashed extracted features are compared to the hashed extracted features of the works to be identified as illustrated in Figure 3 of Conwell using a lookup table. The 3-digit numbers on the left side of Figure 3 (*e.g.*, "034 112 198") are examples of decoded hashed identifiers of the records in the database ("decod[ed] identifier from audio content") that is compared with the hashed identifier of a work to be identified. Conwell Figure 3; 3:46-50.

339. Petitioner's Declarant, Dr. Moulin, confirmed that Petitioner's anticipation theory is based on comparing the hashed extracted features of the work to be identified with the hashed extracted features of the reference works (this comparing is labeled ❸ in the diagram above):

Q We discussed earlier that in your analysis of Conwell in anticipation, you identified it as the extracted features of the first electronic work, the hash of the unknown work; right?
A Let me just read it again.
Yes. So the feature is the hash that's extracted in these.
Q And we compare that with the first electronic data in the database; right?
A Yes.
Q That is the hash of the reference works; right?
A Yes.

Moulin Depo. 263:22-264:9;

Is it your assertion that the nonexhaustive neighbor search here is the search that's done when we use the hash value lookup table?
A Yes. So if you have computed -- say our identifier was 198 in decimal representation, we simply have to look up, is that number in the left column of the table. And that's a straightforward thing to do; so the search is trivial.

Moulin Depo. 199:23-200:5; 191:8-12; 173:25-174:6; 170:14-18; 185:20-24.

Is it your testimony that Conwell teaches Element (c) when it teaches using a robust hash approach, where it uses a lookup table to compare one hash value of the unknown work to hash values that are in the database?
A Yes.

Moulin Depo. 194:10-15.

340. Extracting the features is part of a "preprocessing step" rather than comparing the features (again, labeled ❸ in the diagram above):

> A. As I said -- okay. You start from the work, let's say audio. You extract features. That is a representation. Okay? That's a preprocessing step. You retain the features. Then you extract a hash, which is a compacted electronic representation. It's bits.

Moulin Depo. 169:4-13.

341. Conwell exclusively teaches "comparing" the hashed extracted features using a "lookup table," which uses an exact match comparison rather than a "neighbor search." Conwell teaches that the hashed extracted features of the reference works and the work to be identified are compared using a "lookup table." Figure 3; Conwell, 3:43-62 ("Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table.") A device decodes a hashed "identifier from audio content" to be identified which is then compared to the hashed identifiers in the lookup table. Conwell, 3:43-62. If the decoded identifier of the work to be identified matches one of the decoded identifiers in the lookup table, "the user's web browser is then directed to that URL." Conwell, 3:43-62.

The "lookup table" disclosed in Conwell used to compare the hashed extracted features of the reference works and the work to be identified uses an exact search rather than a neighbor search.

342. The "lookup table" disclosed in Conwell looks for an exact match of the identifiers. If there is an exact match, the search identifies that match. If there is a "neighbor"—*i.e.*, "a close, but not necessarily exact or closest, match," the search will not identify such a neighbor. A neighbor search—a search that identifies "a closes, but not necessarily exact or closest, match" (*see* Section V(C))—can identify an exact match; however, it must also be able to identify a close match. The lookup table search in Conwell cannot identify a record whose has value is a close match. If the hashed identifier of the work to be identified does not have an exact match with the hashed identifiers in the reference database, the result will not be an exact match and no match will be identified even if there is a close or closest match. For example, as illustrated in Figure 3 of Conwell, if the identifier of the work to be identified is 199, it will not result in a match even though it is very close to 198, a match for the URL www.supertracks.com... in Figure 3. My understanding is confirmed by Petitioner's Declarant:

Q. In order to do this approach, are we going
to have to take our unknown work and also hash it to
create a 128-bit hash value?
A. Yes. And so as an example, if that
128-bit string corresponds to the number 125, then
the lookup table tells us, "Okay. This is
www.supertracks," and so on. But if it was 126, we
simply don't find it.

Moulin Depo. 188:13-20.

343. Petitioner's Declarant confirmed my understanding—that comparing the hashed extracted features of the reference database with the hashed extracted features of the work to be identified "will never" identify a "neighbor"—that is, it is will never identify "a close, but not necessarily exact or closest, match:"

Q. If -- in Conwell, when it does a
comparison of the extracted features of the first
electronic work with the first electronic data, it's
comparing a hash value with a set of hash values;
right?
A. Yes.
Q. That comparison is always going to produce
either an exact match or no match; right?
A. Yes.
Q. That -- that comparison will never return
a suggestion that, "Here's something that doesn't
quite match, but it's close"?
A. That's correct.

Moulin Depo. 264:13-25.

> Q All right. In Conwell, if when we start with -- on the one hand, we start with a hash value of a -- of an unknown work; right? Yes?
> A Okay. Fine.
> Q And we're going to compare that to a table of hash values of reference works; right?
> A Yes.
> Q Is that -- is the case that we're always going to return either an exact value, an exact match, or no match?
> A In terms of the identifier, yes, it is true.

Moulin Depo. 259:20-260:6.[35] This exact match lookup table is the only search disclosed in Conwell. Conwell, 3:43-62. Petitioner's Declarant again confirmed my understanding of Cowell:

> So the only search that's disclosed in -- in Conwell using the hash values is one where we look for an exact match, and it's either there or it isn't; is that right?
> A Well, an exact match of that identifier. Like 198, is it there or not? If I have 199, I would say it's not there.

Moulin Depo. 201:4-10; 199:18-21.

---

[35] As I explained above, the "identifier" is what is actually compared using the search disclosed in Conwell.

344. Accordingly, as confirmed by Petitioner's Declarant, the lookup search disclosed in Conwell is not a neighbor search because the hashed identifiers disclosed in Conwell are never used to perform a neighbor search as required by each independent claim of the '179 patent.

> Q. If we look at the search you've identified as constituting the neighbor search, it's a search wherein we start with a hash value and then go to a lookup table to see if it appears in that table; is that right?
>
> A. Yes.
>
> Q. Is the result of that going to be either an exact match of the hash or a determination that it does not exist?
>
> A. Yeah. It's either the table or it's not.
>
> Q. Are we ever going to have a circumstance where we look in the table for that hash value and we conclude that the hash value does not appear, but here's one that's pretty close, and we'll return that one?
>
> A. It's not -- it does not -- Conwell does not disclose that.

Moulin Depo. 200:11-201:2;

Q I want you to assume that we define a neighbor search as one that produces a close but not necessarily exact match, so that a search that always produces an exact match would not be a neighbor search.
With that definition, if we have a search that consists of a hash value lookup in a table of reference hash values, is that lookup a neighbor search?
MR. ELACQUA: Objection.
THE WITNESS: Under your flawed definition, it will not be a neighbor search.
BY MR. DOVEL:
Q Why not?
A By your own definition.
Q What about my definition would make that not a neighbor search?
A Well, because you said if there's an exact match always, you defined that as it's not a neighbor search. And therefore, in the example you said with hash -- with the lookup table, it will not be a neighbor search according to your own definition.
Q How doesn't it meet the definition?
A Because you declared that a search that always produces a -- an exact match for some reason, you declared that not to be a neighbor search.
Q And does a hash value always -- hash lookup always produce an exact match?
A Or no match.

Moulin Depo. 262:10-263:14.

345. In addition to not being expressly disclosed in Conwell, "comparing [the extracted features] using [a] neighbor search" is not inherent. I understanding that a claim limitation is inherent in the prior art if it is necessarily present in the prior art, not merely probably or possibly present." Neither the Petition nor Petitioner's Declaration states (or even suggests) that such a search is inherent or necessarily present in Conwell. *See, e.g.*, Moulin Depo. 305:22-25:

> Q. you're all expressing any opinion to
> the effect that a element was not expressly taught
> but was instead inherent?
> A. I don't recall making that statement, no.

346. While the search disclosed in Conwell may have the result of identifying a neighbor of the work to be identified, it does not do so by "comparing [the extracted features] using [a] neighbor search" as required by the '179 claims. It is my understanding that if a process disclosed in a reference achieves the same result as a claimed invention, the reference does not anticipate the claimed process unless the disclosed process achieves that same result using the claimed steps, as illustrated by the following simple analogy. Assume (a) the claims require filing a Patent Owner's Response at the Patent using e-filing; (b) a reference discloses a process of filing Patent Owner Responses at the Patent Office, but teaches doing so by hand filing. Even though the reference has the same result—a Patent Owner Response is filed at the Patent Office—the reference does not anticipate because it

achieves the same result using a different process than the claimed invention. Petitioner's Declarant agreed with my understanding that different processes can achieve the same results exists in the context of searching:

> Q Well, you would agree that sometimes you
> can reach the same result using different processes;
> right? There's all different kinds of search
> processes; right?
> A Yes.

Moulin Depo. 266:11-15.

> Q You might be able to have a certain
> process that would result in identifying neighbors,
> but it wouldn't use a neighbor search; right?
> A That's possible.

Moulin Depo. 266:21-24.

347. As I described above, had the system disclosed in Conwell been able to identify a neighbor of an underlying work, it would do so using a different process than the process of comparing of extracted features using a neighbor search as claimed in the '179 Patent. Petitioner's expert confirmed that his analysis is based on comparing the "feature space" rather than comparing the extracted features using a neighbor search as required by the claim language. He also confirmed that comparing the extracted features as required by the claims results in an exact match search – there's a match or, no:

Q. Is it the case that if "neighbor search"
is defined as a search that produces a close but not
necessarily exact match, then that a search that
always produces an exact match is not a neighbor
search, would a hash lookup that compares the hash
value to a set of hash values in a table be a
neighbor search.

MR. ELATUA: Objection.

THE WITNESS: I disagree with the definition.

The reality is extremely simple. You --
the neighborhood is defined in the feature space.
You have similar features. If they map to the same
identifier, you -- that defines your search
algorithm. It simply looks at the table, and it
says, "Yes, here's a match," or "no."

Moulin Depo. 260:7-21.

348. Had the search process taught in Conwell identified a neighbor of the work to be identified (*i.e.*, achieves the same result as the '179 claims), it would still not anticipate the '179 claims because it achieves that result using a different process—*i.e.*, using an exact match comparison of hashed extracted features.

349. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches "comparing [the extracted features] using [a] neighbor search."

350. Petition: As support that Conwell discloses the claimed "comparing [the extracted features] using [a] neighbor search," the Petition relies on the

highlighted sentence in the following paragraph (and the corresponding citations to Conwell and the Declaration):

> Third, the '179 Claims require "identifying" the unknown work "using a non-exhaustive neighbor search." Ex. 1001 at Claims 1, 13, 25. Conwell discloses identifying a work by performing a lookup in a hash table. Ex. 1009 at 3:43-62, Figs. 1, 3. Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar "neighbor" results even when the input work is not identical to the reference work. *Id.* at 4:64-5:3; Ex. 1004 at ¶ 86. This search is non-exhaustive because it does not require a

Pet. ('179) at 24.

351. Petition chart: The chart in the Petition makes the same assertion and cites the same passage:

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Conwell identifies an electronic work by accessing a lookup table in a computer system to "decode an identifier [i.e., extracted features] from the audio content and send the identifier to the database, [which] responds by returning the URL corresponding to that identifier back to the user device." 3:43-62, Figs. 3-4 This is a neighbor search because "non-identical versions of the same basic content may nonetheless correspond to the same identifier." 4:64-5:3; Ex. 1004 at ¶ 86. This search is non-exhaustive because it uses a sorted lookup table that uses work identifiers as an index. 3:43-50, 5:58-64; Ex. 1004 at ¶ 86.

Pet. ('179) at 25-26.

352. Declaration: Petitioner's Declarant makes the same assertion and cites the same passage:

> 86. Third, it is my opinion that Conwell discloses "identifying" the unknown work "using a non-exhaustive neighbor search." Ex. 1001 at Claims 1, 13, 25. Conwell teaches this limitation in the form of a table look-up using the identifier decoded from the electronic work. *See* Ex. 1009 at 3:43-62, Figs. 1, 3. Once the user device decodes the identifier from the work, it is sent to the database. *Id.* at 3:46-50. It is my opinion that this constitutes a neighbor search. Specifically, because Conwell also teaches the use of a hash algorithm such that similar, but non-identical, inputs generate the same output identifier, the table look-up will return similar "neighbor" results even when the input work is not identical to the reference work. *See id.* at 4:64-5:3. Further, it is my opinion that

Moulin Decl. ('179) ¶86

353. Declarant chart: Finally, the chart in Petitioner's Declaration makes the same assertion and cites the same passage:

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Conwell teaches identifying a first electronic work by accessing a lookup table in a computer system to "decode an identifier [i.e., extracted features] from the audio content and send the identifier to the database, [which] responds by returning the URL corresponding to that identifier back to the user device." 3:43-62, Figs. 3-4. Conwell further teaches a neighbor search because "non-identical versions of the same basic content may nonetheless correspond to the same identifier." 4:64-5:3. Conwell further discloses that this neighbor search is non-exhaustive under Petitioner's construction because it uses a sorted lookup table that uses work identifiers as an index. 3:43-50, 5:58-64

Moulin Decl. ('179) ¶89.

354. The Petition, Declaration, and corresponding charts fail to establish that Conwell teaches the claimed "comparing [the extracted features] using [a] neighbor search." In fact, the Petition, Declaration, and charts fail to address the actual claim language. As set forth above, the claims require

> work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;

'179 claim 1, element [c]. I note that this claim language and concept is completely absent from Petitioner's analysis. The Petition and Declaration fail to address what is actually being compared using the neighbor search.

355. The only comparing identified in the petition is "performing a lookup in a hash table" (Pet. ('179) at 24), which, as demonstrated above, is an exact match comparison, not a neighbor search. Instead, the Petition and Declaration state that "the '179 Claims require identifying the unknown work 'using a non-exhaustive neighbor search.'" Pet. ('179) at 24. This mischaracterization of the claims ignores the actual claim language. As set forth above, the claims do not simply require "'identifying' the unknown work 'using a non-exhaustive neighbor search;'" rather, they require:

> work by comparing the extracted features of the first electronic work with the first electronic data in the data-base using a non-exhaustive neighbor search;

('179 claim 1, element [c]), and the only comparing of extracted features identified in the Petition (and disclosed in Conwell) is an exact match comparison—"performing a lookup in a hash table." (Pet. at 24).

356. Petitioner observes that the system disclosed in Conwell may achieve the same result as the result of the '179 claims:

> "Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar 'neighbor' results even when the input work is not identical to the reference work."

Pet. ('179) at 24 (citing Conwell, 4:64-5:3).[36] But, as I noted above, achieving this same result using a process that is different than that claimed in the '179 claims does not anticipate the '179 claims. For example, the Declarant states that "because Conwell also teaches the use of a hash algorithm such that similar, but non-identical inputs generate the same output identifier, the table look-up will return similar 'neighbor' results..." Moulin Decl. ('179) ¶86. That the search generates similar "neighbor" results, however, does not address the claim language and, in particular, what is actually being compared using the claimed "neighbor search."

357. The two passages from Conwell cited in the Petition and Declaration (the first attempting to establish "identifying" in general and the second attempting to establish the "neighbor search") do not demonstrate the claimed "comparing [the extracted features] using [a] neighbor search." I address each passage in turn:

358. Conwell, 3:43-62:

[36] Conwell does not identify an example of such a hashing algorithm.

> Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier from audio content and send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device. The user device then directs an internet browser to that URL. By such arrangements, music (e.g., in MP3 format) can serve as a portal to a web site dedicated to the music artist, a web site giving concert schedules for the artist, a web site offering CDs, etc.
>
> In the FIG. 3 example, if the device decodes the identifier '376' from an MP3 file, and queries the database with this data, the database returns the URL www.emusic.com/0555353x.pdf. The user's web browser is then directed to that URL. (For expository convenience, the identifiers are assumed to be in the range 0–1023. In actual implementations, a much larger range would usually be used.)

One skilled in the art would understand that this passage identifies the exact match "look-up table" illustrated in Figure 3 which, as described above, is an exact match search rather than a neighbor search. This passage does not teach (and the Petition and Declaration do not contend that it teaches) "comparing [the extracted features] using [a] neighbor search."

359. Conwell, 4:64-5:3:

> Of course, by suitably designing the algorithm by which identifiers are derived, non-identical versions of the same basic content may nonetheless correspond to the same identifier. There is extensive published research on such technology, e.g., hashing algorithms by which similar or related, but non-identical, inputs map to the same hash outputs.

One skilled in the art would understand that this cited passage states that similar works might correspond to the same hash. Using Figure 3 to illustrate, this passage suggests that two similar works might produce the same hash, e.g., the 4th entry in the table "376." The passage does not state that the hash identifier (*e.g.*, extracted feature 376) is then compared to the hash of the reference work using the claimed "neighbor search." Rather, as confirmed by the prior passage, the hash identifier (extracted feature) is compared only using an exact match lookup table.

360. *The Board's concerns*: I now address the Board's specific concerns (identified in its Decision in the '179 IPR) with respect to whether Conwell discloses a "neighbor search." In instituting Ground 1, the Board recognized that Conwell discloses a search that compares the hashed output identifier (*i.e.*, "digitally created compact electronic representation of the first electronic work") with the records in the database using an "exact match lookup" but instituted Ground 1 because the "user would be directed to the same URL for both Song A and Song A1, thereby matching both Song A and Song A1":

> In contrast to Levy, Conwell discloses that its hash algorithm generates the same output identifier for similar, but non-identical inputs, as discussed by Petitioner. Pet. 24 (citing Ex. 1009, 4:64–5:3). Patent Owner acknowledges the citation, but argues that the same "hashed output identifier is compared to the records using an exact match lookup." Prelim. Resp. 29. Patent Owner continues that such an exact match search is not the same as a "neighbor" search recited in the claims. We are persuaded, however, that similar, but non-identical, inputs, Song A and Song A_1, in Patent Owner's example (*id.* at 28), would be identified and the same URL provided by the hash table lookup in Conwell. In other words, the search, or lookup, would "identify[] a close, but not necessarily exact or closest, match," such that a user would be directed to the same URL for both Song A and Song A_1, thereby matching both Song A and Song A_1. Although it appears that such a search process is different from that described in the Specification of the '179 Patent, we are persuaded, on the present record, that it meets the discussed limitations recited in the independent claims.

Decision ('179) at 11-12.

361. As I demonstrated above, the search process disclosed in Conwell is not only "different from that described in the Specification of the '179 Patent" (Decision ('179) at 12), it is also different than the process claimed in each independent claim of the '179 Patent. As I explained above, each independent claim requires "comparing [the extracted features] using [a] neighbor search:"

> work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;

'179 claim 1, element [c]. Conwell exclusively discloses doing such comparing using an exact match lookup table. Conwell, 3:43-62; Moulin Depo. 200-11-201:2.

362. One skilled in the art would understand that the Board's analysis addresses alternative claim language that does not require comparing the extracted features using a neighbor search but instead simply identifies a neighbor (*e.g.*, Song A1) independent of the actual search being performed. That the system disclosed in Conwell may achieve the same result—*i.e.*, map both Song A and Song A1 to the same identifier—does not convert the exact match comparison of the hashed identifiers disclosed in Conwell into a neighbor search of such identifiers as required by each independent claim.

2. non-exhaustive search (claims 1, 13, 25).

363. In instituting Ground 1, I note that the Board did not specifically address whether Conwell discloses the claimed non-exhaustive search. Decision ('179) at 11-12. The search disclosed in Conwell is not nonexhaustive.

364. As I explained above, the claimed "non-exhaustive" search is "a search that locates a match without a comparison of all possible matches." Section V(B); Decision ('179) at 7. A non-exhaustive search uses an intelligent algorithm to reduce the number of potential matches. By contrast, an exhaustive search uses brute force to compare the work to be identified with each record in the database,

"perhaps halting the search when the first match is found." '237, 8:59-61; *see* Section V(B).

365. Conwell does not teach the claimed "non-exhaustive …search." As illustrated in Figure 3, Conwell teaches identifying a match using "a large look-up table." Conwell, 3:43-44 ("Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table."). Conwell also discloses (in a section addressing how the table can be maintained) that the "look-up table" can include "entries … sorted, by identifier." Conwell, 5:59-61. These disclosures in Conwell are neither an expressed nor an inherent disclosure of a "non-exhaustive search."

366. Express: Conwell does not expressly disclose using the "look-up table" to conduct a non-exhaustive search—*i.e.*, using an algorithm that increases efficiency by intelligently searching only a subset of potential matches rather than a "brute force" search. There are many potential ways to use Conwell's "look-up table" to identify a match. For example, one possible approach would be to compare the hashed identifier of the extracted features of the work to be matched with the first entry in the look-up table, then with the second entry, and so on—*i.e.*, an exhaustive search. Using Figure 3 as an illustrative example, if the hashed identifier of the work to be identified is 612, the identifier could be compared with the first entry 034, resulting in no match; the search would then compare the

identifier to the second entry in the lookup table 112, again resulting in no match; and so on until the search reached 612—a match.

034	www.sonymusic.com/catalog/05634.html
112	www.sonymusic.com/catalog/00014.html
198	www.superdtracks.com/index/artists/taylor.htm
376	www.emusic.com/0555353x.pdf
597	www.cdw.com/music/featured_CDs/index.html
612	www.sonymusic.com/catalog/00231.html
850	www.polygram.com/franklin/adf_234.htm
921	www.loudeye.com/rap/1999/46755646.html

FIG. 3

Conwell, Figure 3.

367. As I explained above, such a search would be exhaustive rather than the claimed non-exhaustive because it systematically checks whether each potential match matches the work to be identified until a match is found. Accordingly, this search would be exhaustive (rather than non-exhaustive) whether or not the searched stopped after identifying a match. An exhaustive search, which uses brute force to compare the work to be identified to reach record without reducing the potential record candidates, can "halt the search when the first match is found." '179, 9:8-13. An exhaustive search systematically checks whether each potential match matches the work to be identified until a match is found, "perhaps halting the search when the first match is found." '179, 9:8-13.

368. As another example, if the hashed identifier of the work to be identified were 744, the identifier could be compared with the first entry 034, resulting in no match; the search would then compare the identifier to the second entry 112, again resulting in no match; and so on until the entire table is compared. Because there is no exact match, no match would be identified. This approach of using the disclosed sorted lookup table does not use a non-exhaustive search, because its searches all entries until a match is found, rather than using an algorithm that increases efficiency by intelligently searching only a subset of potential matches.

369. While there are ways to search the lookup table disclosed in Conwell using a non-exhaustive approach, Conwell does not disclose any such non-exhaustive approach. In fact, Conwell does not teach any specific method of conducting the exact match comparison using the disclosed lookup table. As I explained above, the only descriptions in Conwell regarding the search to be used are the following generic statements: (1) the "database responds…."—which does not disclose a non-exhaustive search; and (2) "queries the database" which also does not disclose a non-exhaustive search:

> Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier from audio content and send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device. The user device then directs an internet browser to that URL. By such arrangements, music (e.g., in MP3 format) can serve as a portal to a web site dedicated to the music artist, a web site giving concert schedules for the artist, a web site offering CDs, etc.
>
> In the FIG. 3 example, if the device decodes the identifier '376' from an MP3 file, and queries the database with this data, the database returns the URL www.emusic.com/0555353x.pdf. The user's web browser is then directed to that URL. (For expository convenience, the identifiers are assumed to be in the range 0-1023. In actual implementations, a much larger range would usually be used.)

Conwell, 3:43-62.

370. Inherent: Conwell also does not inherently disclose using the "look-up table" to conduct a non-exhaustive search. First, as a preliminary matter, I note that neither the Petition nor the Petitioner's Declarant relied on any theory that Conwell inherently discloses a non-exhaustive search. Pet. ('179) 1-60. Accordingly, because the Petitioner did not present this theory, it is my understanding that it cannot be a basis for finding the '179 claims unpatentable.

371. Second, Conwell does not inherently disclose a "non-exhaustive search." It is my understanding that a claim limitation is inherent in the prior art if it is necessarily present in the prior art, not merely probably or possibly present. As I noted above, Conwell does not expressly disclose comparing the hashed

identifiers to the potential matches in the look-up table (sorted or unsorted) using a non-exhaustive search. And, as illustrated in the example above, such a search is not "necessarily present"—there are many ways to compare a hashed identifier to the reference hashed identifiers in a look-up table other than using a non-exhaustive search. While it might be possible or maybe even probable that one skilled in the art could use a non-exhaustive approach to conduct a search using the look-up table disclosed in Conwell, such a possibility or even probability does not establish an inherent disclosure.

372. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches "comparing [the extracted features] using a non-exhaustive… search."

373. Petition: As support that Conwell discloses the claimed "comparing [the extracted features] using a non-exhaustive … search," the Petition relies on the following:

> This search is non-exhaustive because it does not require a brute force comparison of all possible hashes, but rather requires a single lookup in a numerically sorted lookup table that uses hash identifiers as an index. *E.g.*, Ex. 1009 at 3:43-50, 5:58-64; Ex. 1004 at ¶ 86.

Pet. ('179) at 24.

374. Petition chart: The chart in the Petition makes the same assertion and cites the same two passages from Conwell:

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Conwell identifies an electronic work by accessing a lookup table in a computer system to "decode an identifier [i.e., extracted features] from the audio content and send the identifier to the database, [which] responds by returning the URL corresponding to that identifier back to the user device." 3:43-62, Figs 3-4. This is a neighbor search because "non-identical versions of the same basic content may nonetheless correspond to the same identifier." 4:64-5:3; Ex. 1004 at ¶ 86. This search is non-exhaustive because it uses a sorted lookup table that uses work identifiers as an index. 3:43-50, 5:58-64; Ex. 1004 at ¶ 86.

Pet. ('179) at 25-26.

375. Declaration: Petitioner's Declarant makes the same assertion and relies on the same two passages from Conwell:

> Further, it is my opinion that this constitutes a non-exhaustive search under Petitioner's construction because it does not require a brute force comparison of all possible hashes, but rather requires a single lookup in a numerically sorted lookup table that uses hash identifiers as an index. *E.g.*, 3:43-50 ("Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL . . ."), 5:58-64 ("the present disclosure assumes that the entries are sorted, by identifier").

Moulin Decl. ('179) ¶86.

376. Declarant chart: And finally, the Declarant's chart also makes the same assertion and cites the same two passages from Conwell:

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search.	Conwell teaches identifying a first electronic work by accessing a lookup table in a computer system to "decode an identifier [i.e., extracted features] from the audio content and send the identifier to the database, [which] responds by returning the URL corresponding to that identifier back to the user device." 3:43-62, Figs. 3-4. Conwell further teaches a neighbor search because "non-identical versions of the same basic content may nonetheless correspond to the same identifier." 4:64-5:3. Conwell further discloses that this neighbor search is non-exhaustive under Petitioner's construction because it uses a sorted lookup table that uses work identifiers as an index. 3:43-50, 5:58-64.

Moulin Decl. ('179) ¶89.

377. Petitioner asserts that Conwell discloses a non-exhaustive search, because using the "lookup-table" "does not require a brute force comparison of all possible hashes," but instead "requires a single lookup in a numerically sorted lookup table." Pet. ('179) at 24. As I explained above, Conwell does not teach any particular search algorithm for searching the "lookup-table" and certainly does not disclose a search algorithm that is not a brute force algorithm or an algorithm that requires a "single lookup." Conwell, 3:43-62. Contrary to Petitioner's assertion, Conwell does not teach using a "single lookup" to identify a work.

378. As I explained above, Conwell provides no details as to how the exact-match search between the hashed identifier of the work to be identified and

the hashed identifiers in the reference database is actually performed. The entire description of the exact match search is these two terse highlighted portions:

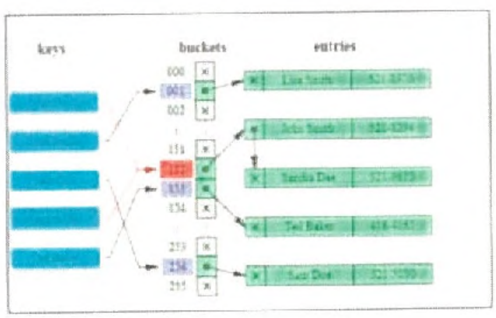
> Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier from audio content and send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device. The user device then directs an internet browser to that URL. By such arrangements, music (e.g., in MP3 format) can serve as a portal to a web site dedicated to the music artist, a web site giving concert schedules for the artist, a web site offering CDs, etc.
>
> In the FIG. 3 example, if the device decodes the identifier '376' from an MP3 file, and queries the database with this data, the database returns the URL www.emusic.com/0555353x.pdf. The user's web browser is then directed to that URL. (For expository convenience, the identifiers are assumed to be in the range 0–1023. In actual implementations, a much larger range would usually be used.)

Conwell, 3:43-62. Neither of these references—"the database responds" or "queries the database"—discloses any particular exact match search, much less a "single lookup" search.

379. Moreover, the illustrative example disclosed in Conwell cannot be used to identify a match using a "single lookup." Hashes can be stored in a standard database structure using [1] just records for the identified keys (as in Figure 3 of Conwell), or [2] pre-allocating records for all possible keys (often referred to as a hash table).

380. The second approach—the hash table—could possibly be used to identify a match using a "single lookup." For example, assume that a hash algorithm generates values from 1-10 and that the reference works in the reference library hash to 1, 4, 7, and 10. A hash table that could be used to perform a "single lookup" would look like this:

Key	URL
1	www....
null	null
null	null
4	www...
null	null
null	null
7	www...
null	null
null	null
10	www....



https://en.wikipedia.org/wiki/Hash_table.[37] The hash table can be looked up directly because there is a line item for every possible number. For example, if the query hashes to a value of 5, the search could go directly to the 5th record.[38]

381. Figures 3 and 4 of Conwell, however, do not have this hash table structure but instead disclose a lookup table that includes gaps (*e.g.*, gaps between 034 and 112, 112 and 198, and 198 and 376)[39]:

---

[37] Each record has a key, but the fields for the work and the associated action will be filled with nulls if no reference work hashed to that lookup value (key/identifier/bucket #).

[38] Because hash tables generally have to accommodate some hash collisions (*i.e.* different files that hash to the same value because there is a limited range of hash values possible), the hash table often has a linked list attached to each hash value bucket that must be traversed to find the actual result. For example, if there are three reference files that all hash to 5, then the query identified above will have to evaluate all three as part of its "single lookup."

[39] The hash value is not the same as the record #; so the search algorithm must search through the records to match the key.

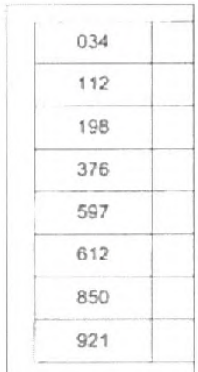


FIG. 3

As a result, the "lookup table" disclosed in Conwell cannot be used to identify a match using a "single lookup" as suggested by Petitioner. Because the records in the table are stored in consecutive positions (even if the hash keys of two adjacent records do not differ by one), we cannot locate the row within that table that stores a particular hash key in a single lookup operation.

382. The two cited passages from Conwell relied on in the Petition, Declaration, and charts do not disclose any particular search algorithm using the sorted lookup-table, much less an algorithm that is a non-exhaustive search. Each passage from Conwell is addressed in turn.

383: Passage 1:

> Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier from audio content and send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device.

Conwell, 3:43-50. This passage does not disclose the claimed "non-exhaustive search" and does not disclose a "single lookup." Rather, this passage states that the "Registry database can be conceptualized as a large look-up table" and states that the "database responds…." The passage says nothing about how the "Registry database" is searched, and specifically does not disclose a "non-exhaustive search" or a "single lookup" as suggested in the Petitioner. Rather, the passage simply states that, when a consumer uses a device to "send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device." Conwell, 3:43-50. The passage says nothing about the search process that uses the "conceptualized … large look-up table" to identify the corresponding URL.

384. <u>Passage 2</u>:

> The maintenance of the table 12 is well understood by those skilled in data structures. For ease of description, the present disclosure assumes that the entries are sorted, by identifier. In actual implementation, this may not be the case. The system may be keyed by identifier, song and artist, thus increasing the speed at which the system can find duplicate songs with different identifiers.

Conwell, 5:58-64. Just like the first passage, this passage also does not disclose the claimed "non-exhaustive search." Rather, this passage states that the (1) the maintenance of the table 12 is well understood by those skilled in data structures, (2) the entries can be sorted by identifier, and (3) the system may be keyed by identifier, sound, and artist.

385. None of these disclose how the search is actually performed, much less that the undisclosed search technique is non-exhaustive. The first three sentences simply describe how the table 12 is maintained and say nothing about how the table is searched. The final sentence addresses how the system is "keyed by identifier, song, and artist." Again this sentence says nothing about how the table is actually searched but rather refers to how the database is maintained.

B. '179 Ground 2: The instituted claims of the '179 Patent are not obvious in view of Ghias and Philyaw.

386. I understand that if a combination of two references fails to teach an important claimed element, it is not possible for that combination to render the claim obvious. That is, assuming one of ordinary skill would have thought to combine the references, that combination would still be missing an important element and therefore, even with the combination, one of ordinary skill would still not possess the invention.

387. All independent claims of the '179 Patent include the "non-exhaustive neighbor search" limitation. '179, claims 1, 13, and 25. In this combination, Petitioner relies exclusively on Ghias rather than Philyaw for the clamed "non-exhaustive neighbor search." Pet. ('179) 50-60. This is confirmed by Petitioner's Declarant:

> Q Did you identify in Philyaw a disclosure of a nonexhaustive neighbor search?
> A I don't recall that, no.

Moulin Depo. 375:18-20;

> Q Is it the case that you only identify Ghias as a reference disclosing a nonexhaustive neighbor search in your combination of Ghias and Philyaw?
> A That's right. Philyaw is not mentioned in that box.

Moulin Depo. 373:23-374:3; Moulin Depo. 374:20-25. Ground 2 fails because Ghias does not disclose the either (1) a non-exhaustive search, or (2) a neighbor search. I address each deficiency in turn.

1. non-exhaustive search (claims 1, 13, 25).

388. The search disclosed in Ghias is not a non-exhaustive search. As I explained above in detail (Section V(B)), a "non-exhaustive…search" is "a search that locates a match without a comparison of all possible matches." Section

V(B); Decision ('179) at 7. Also as I explained above in detail (Section VII(A)(1)(b)), Ghias teaches an exhaustive search that compares the work to be identified (user input 23) with "all the songs" in the database—*i.e.*, "all possible matches." I note that Petitioner's Declarant repeatedly confirmed that the search disclosed in Ghias compares the song to be identified with each record in the database and is therefore non-exhaustive—"a search that locates a match without a comparison of all possible matches." *See* Section VII(A)(1)(b); Moulin Depo. 327:3-12; 327:14-328:4.

389. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a non-exhaustive search.

390. <u>Petition</u>: As support for the claimed "non-exhaustive … search," the Petition relies on the following two quotes from Ghias (labeled (1) and (2)):

> Ghias further discloses that this search may be non-exhaustive. (1) "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. *Id.* at 6:7-11, 6:23-35 (2) ("Several Algorithms have been developed … Running times have <u>ranged from</u> O(mn) for <u>the brute force algorithm</u> <u>to</u> O(kn) or O(nlog(m)").

Pet. ('179) 47-48.

391. <u>Petition Chart</u>: The chart in the Petition cites to these same two passages from Ghias:

'179 patent	Ghias (Ex. 1010) and Philyaw (Ex. 1014)
(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search.	Ghias discloses searching the melody database 14 to identify a matching melody. 2:56-59; Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences matching the melody."). This search may be non-exhaustive: (1) "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. 6:7-11; 6:23-35 ("Several Algorithms have been (2) developed . . . Running times have <u>ranged from</u> O(mn) for <u>the brute force algorithm</u> <u>to</u> O(kn) or O(nlog(m)"). Ghias further discloses that this is a neighbor search that determines [illegible]

Pet. ('179) at 51.

392. <u>Declaration</u>: Petitioner's Declarant addresses these same two passages:

> Ghias further discloses that this search may be non-exhaustive. Specifically, Ghias teaches that "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. Ex. 1010 at 6:7-11. Moreover, Ghias teaches that "Several Algorithms have been developed that address [this] problem" ranging from "brute force" to substantially faster algorithms. *Id.* at 6:23-35 ("Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from O(mn) for the brute force algorithm to O(kn) or O(nlog(m)), where 'O' means 'on the order of,' m is the number of pitch differences in the query, and n is the size of the string (song).")

Moulin Decl. ('179) ¶120.

393. Declarant's Chart: Finally, the Declarant's chart also relies on these same two passages:

(e) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	[illegible] Ghias further discloses that this search may be non-exhaustive. "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. 6:7-11. (1) Moreover, Ghias teaches that "[s]everal Algorithms have been developed that address the problem of approximate string matching. Running times have <u>ranged from</u> O(mn) for <u>the brute force algorithm to</u> O(kn) or O(nlog(m)), where 'O' means 'on the order of,' m is the number of pitch differences in the query, and n is the size of the string (song).") 6:23-35. (2) [illegible]

Moulin Decl. ('179) ¶129.

394. Petitioner and Declarant made these same arguments and pointed to these same two quotes from Ghias trying to establish this same non-exhaustive search element for the '237 Patent. As I explained in detail above with respect to the '237 Patent, Petitioner's assertion and the two passages from Ghias do not disclose the claimed non-exhaustive search.

395. <u>*The Board's Concerns*</u>: I now address the Board's specific concerns with respect to whether Ghias discloses the claimed non-exhaustive search. In instituting Ground 2, the Board did not rely on the arguments presented by Petitioner and its Declarant or the passages from Ghias quoted by Petitioner and its

Declarant attempting to establish the claimed non-exhaustive search. Instead, the Board preliminary found that Ghias disclosed the "non-exhaustive" search because the search disclosed in Ghias could produce a list of matches based on an error-tolerance and the user can perform a "new query on a restricted search list consisting of songs just retrieved":

> Ghias provides that "[t]he number of matches that the database 14 should retrieve depends upon the *error-tolerance* used during the key-search," and "the user can perform *a new query on a restricted search list consisting of songs just retrieved*. This allows the user to identify sets of songs that contain similar melodies." Ex. 1010, 6:63–65, 7:5–8 (emphases added). Thus, Ghias makes clear that the search need not be exhaustive, as Patent Owner has argued, and will act to "identify[] a close, but not necessarily exact or closest, match," per our claim construction.

Decision ('179) at 14.

396. As I explained above in detail, there are two reasons why the Board's reliance on the "new query on a restricted search list" does not satisfy Petitioner's burden of demonstrating that the instituted clams of the '179 Patent are untenable based on Ghias.

Reason 1: Had these passages from Ghias cited by the Board disclosed the claimed non-exhaustive search (they do not), it is my understanding that it would be improper for the Board to rely on these passages to find '179 claims unpatentable because the passages were not identified in the Petition as support for the non-exhaustive search.

Reason 2: As I explained above in detail, using a query on the "restricted search list consisting of songs just received" does not disclose the claimed "non-exhaustive search."

397. The Board also noted that if Ghias disclosed a non-exhaustive search, Ghias would still disclose this claimed element even if Ghias also disclosed other searches (*e.g.*, exhaustive searches):

> Additionally, given the "comprising" language used in the independent claims, we are not persuaded that the claimed methods could not cover processes with both exhaustive and non-exhaustive searching, as long as the latter provides identification.

Decision ('179) at 14. Because, as demonstrated above, Ghias does not disclose any non-exhaustive searching, Ghias does not disclose this claimed element.

2. neighbor search (claims 1, 13, 25).

398. I note that in instituting this Ground, the Board did not specifically address whether Ghias discloses the claimed neighbor search. The search disclosed in Ghias is not a neighbor search.

399. As I explained above in detail (Section V(C)), a "neighbor" search is a search that identifies "a close, but not necessarily exact or closest, match." *See* Section V(C); Decision ('179) at 8. Also as I explained above in detail, Ghias does

not disclose a neighbor search because the search disclosed in Ghias always (necessarily) identifies an exact or the closest match—the disclosed search is guaranteed to find) the closest match. Section VI(B)(2)(b).

400. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a neighbor search.

401. The Petition relies on the following as support for the claimed neighbor search:

> Ghias further discloses that this search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." *Id.* at 2:50-59, 6:60-63.

Pet. ('179) 47-48.

402. The chart in the Petition presents the same reason and quotes the same passages from Ghias:

'179 patent	Ghias (Ex. 1010) and Philyaw (Ex. 1014)
(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Ghias discloses "search[ing] the melody database 14" to identify a matching melody. 2:50-59, Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences . . . match[ing] . . . the melody."). This search may be non-exhaustive: "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. 6:7-11; 6:23-35 ("Several Algorithms have been developed . . . Running times have ranged from O(mn) for the brute force algorithm to O(kn) or O(nlog(m)"). Ghias further discloses that this is a neighbor search that determines "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." 2:50-59, 6:60-63.

Pet. ('179) 51.

403. The Petitioner's Declarant also presents the same reason and cites the same two passages from Ghias:

> 122. Ghias further discloses that this search locates a near or nearest neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." Ex. 1010 at 2:50-59, 6:60-63.

Moulin Decl. ('179) ¶122.

404. Finally, the chart in the Declaration also presents the same reason and cites the same two passages:

(e) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search.	Ghias discloses "search[ing] the melody database 14" to identify a matching melody 2:50-59. Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences which at least approximately matches the sequence of digitized representations of relative pitch differences of the melody.") Ghias further discloses that this search may be non-exhaustive. "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. 6:7-11. Moreover, Ghias teaches that "[s]everal Algorithms have been developed that address the problem of approximate string matching. Running times have <u>ranged from</u> O(mn) for <u>the brute force algorithm</u> <u>to</u> O(kn) or O(nlog(m)), where 'O' means 'on the order of,' m is the number of pitch differences in the query, and n is the size of the string (song).") 6:23-35. Ghias further discloses that this is a neighbor search that determines "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." 2:50-59, 6:60-63.

Moulin Decl. ('179) ¶129.

405. Neither of the cited passages from Ghias discloses the claimed neighbor search because, as described above, searches that produce either the ranked list or single most approximate matching melody always identify the closet match. I address each passage in turn.

406. <u>Passage 1</u>:

> The query engine 24 searches the melody database 14 and outputs a ranked list of approximately matching melodies, as illustrated at 26. A preselected error tolerance may be applied to the search. The query engine 24 may of course alternatively be programmed to output the single most approximate matching melody or, if desired, to output an exact matching melody. However, by searching for an approximate matching melody, as hereinafter discussed, various forms of anticipated errors may be taken into account.

Ghias, 2:50-59. As noted in the Petition and Declaration, this passage states that the search "outputs a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." As I explained above, neither approach discloses the claimed "neighbor search." A "neighbor search" must identify "a close, but not necessarily exact or closest, match." Decision ('170) at 8. Both of the searches disclosed in this passage necessarily disclose an exact or the closest match and, therefore are not "neighbor searches."

407. Passage 2:

> The computer 16 may desirably be programmed so that, for a given query, the database 14 returns a list of songs ranked by how well they matched the query, not just one best match.

Ghias, 6:60-63. This passage does not disclose a neighbor search. As I explained above, the "list of songs ranked by how well they matched the query" necessarily identifies an exact or the closest match, and specifically identifies such song as the top-ranked song.

IX. '441 Patent.

408. The Board instituted the '441 IPR based on the following two grounds:

- Ground 1: Claims 1–3, 6, 8–14, 19, 21–26, and 30 under 35 U.S.C. § 102(e) as anticipated by Conwell; and
- Ground 2: Claims 1–3, 8, 10–14, 18, 19, 21–27, 29, and 30 under 35 U.S.C. § 103(a) as obvious over Ghias and Philyaw;

Decision ('441) at 15. I address each ground in turn.

A. '441 Ground 1: The instituted claims of the '441 Patent are not anticipated by Conwell.

409. Each independent claim of the '441 Patent includes a "non-exhaustive neighbor search" limitation. '441 claims 1, 13, and 25. Ground 1 fails because Conwell does not disclose (1) a non-exhaustive search, and (2) a neighbor search. I address each deficiency in turn.

1. neighbor search (claims 1, 13, 25).

410. Conwell does not disclose the claimed neighbor search.

411. Like the '179 Patent, each independent claim of the '441 Patent requires a very specific comparison—"comparing [the extracted features] using [a] neighbor search"—that is:

- Claim 1: "comparing [1] the extracted features of the first electronic work with [2] the first electronic data in the database using a non-exhaustive neighbor search."
- Claim 13: "comparing [1] the first electronic data with [2] the second digitally created compact electronic representation using a non-exhaustive neighbor search."
- Claim 25: "comparing [1] the first electronic data with [2] the second digitally created compact electronic representation of the first electronic work using a non-exhaustive neighbor search."

412. As I demonstrated above with respect to the '179 Patent,

- a "neighbor search" is a search "identifying a close, but not necessarily exact or closest, match;" and
- Conwell does not teach "comparing [the extracted features] using a … neighbor search." Rather Conwell teaches comparing these features using an exact match lookup table.

413. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches a "neighbor search." As support that Conwell

discloses the claimed "neighbor search" for the '441 Patent, the Petition and corresponding Declaration rely on the same discussion and citations to Conwell that they identified for this claim element with respect to the '179 Patent addressed above. Compare Pet. ('179) at 24 with Pet. ('441) at 23-24; Moulin Decl. ('179) ¶86 with Moulin Decl. ('441) ¶86.

414. In the '179 Petition, Petitioner asserted:

> Third, the '179 Claims require "identifying" the unknown work "using a non-exhaustive neighbor search." Ex. 1001 at Claims 1, 13, 25. Conwell discloses identifying a work by performing a lookup in a hash table. Ex. 1009 at 3:43-62, Figs. 1, 3. Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar "neighbor" results even when the input work is not identical to the reference work. *Id.* at 4:64-5:3; Ex. 1004 at ¶ 86. This search is non-exhaustive because it does not require a

Pet. ('179) at 24. In the '441 Petition, Petitioner made the same assertion:

> Third, the '441 Claims require "identifying . . . the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search." Ex. 1001 at Claims 1, 13, 25. Conwell teaches table look-up using the identifier decoded from the electronic work. *See* Ex. 1009 at 3:43-62, Figs. 1, 3. Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar "neighbor" results even when the input work is not identical to the reference work. *Id.* at 4:64-5:3; Ex. 1004 at ¶ 86. This search is non-

Pet. ('441) 23. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches neighbor search for the same reasons set forth above in detail with respect to this element as used in the '179 Patent.

2. non-exhaustive search (claims 1, 13, 25).

415. In instituting Ground 1, I note that the Board did not specifically address whether Conwell discloses the claimed non-exhaustive search. Decision ('441) at 11-12.

416. Conwell does not disclose the claimed non-exhaustive search. As I explained above in detail in with respect to the '179 Patent,

(1) the claimed "non-exhaustive … search" is a search that "locates a match without a comparison of all possible matches," and

(2) Conwell dose not teach a "non-exhaustive search" that "locates a match without a comparison of all possible matches."

417. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches comparing the extracted features using a "non-exhaustive … search." As support for this element, the Petition, Declaration, and corresponding charts rely on the same discussion and citations to Conwell identified for this claim element with respect to the '179 Patent addressed above.

Compare Pet. ('179) at 24 with Pet. ('441) at 23-24; Moulin Decl. ('179) ¶86 with Moulin Decl. ('441) ¶86.

418. For the '179 Patent, Petitioner asserts:

> This search is non-exhaustive because it does not require a brute force comparison of all possible hashes, but rather requires a single lookup in a numerically sorted lookup table that uses hash identifiers as an index. *E.g.*, Ex. 1009 at 3:43-50, 5:58-64; Ex. 1004 at ¶ 86.

Pet. ('179) at 24.

419. For the '441 Patent, Petitioner makes the same assertion:

> This search is non-exhaustive because it does not require a brute force comparison of all possible hashes, but rather requires a single lookup in a numerically sorted lookup table that uses hash identifiers as an index. Ex. 1009 at 3:43-50, 5:58-64; Ex. 1004 at ¶ 86.

Pet. ('441) at 23-24. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches a non-exhaustive search for the same reasons set forth above in detail with respect to this element as used in the '179 Patent.

B. '441 Ground 2: The instituted claims of the '441 Patent are not obvious over Ghias and Philyaw.

420. It is my understanding that if a combination of two references fails to teach an important claimed element, it is not possible for that combination to

render the claim obvious. That is, assuming one of ordinary skill would have thought to combine prior art references, that combination would still be missing an important claim element and therefore, even with the combination, one of ordinary skill would still not possess the invention.

421. Any combination of Ghias with Philyaw is missing the claimed "non-exhaustive neighbor search."

422. All independent claims of the '441 Patent include the limitation "non-exhaustive neighbor search." '441, claims 1, 13, and 25.

423. For Ground 2, I note that Petitioner again relies exclusively on Ghias for the clamed "non-exhaustive neighbor search." Pet. ('441) at 49-60; Moulin Depo. 375:18-20; 373:23-374:3; 374:20-25. Ground 2 fails because Ghias does not disclose (1) a non-exhaustive search, and (2) a neighbor search. I address each deficiency in turn.

1. non-exhaustive search (claims 1, 13, 25).

424. As I explained above in detail:

(a) a non-exhaustive search is "a search that locates a match without a comparison of all possible matches," and

(b) Ghias teaches an exhaustive search that compares the work to be identified with "all the songs" in the database—i.e., "all possible matches."

425. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a non-exhaustive search. As support for this element, I note that the Petition, Declaration, and corresponding charts rely on the same discussion and citations to Ghias identified for this claim element with respect to the '179 Patent addressed above. Compare Pet. ('179) 47-48 with Pet. ('441) 47-48; Moulin Decl. ('179) ¶¶120-121 with Moulin Decl. ('441) ¶¶120-121.

426. For the '179 patent, Petitioner asserts:

> between successive notes of the melody."). Ghias further discloses that this search (1) may be non-exhaustive: "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. *Id.* at 6:7-11; (2) 6:23-35 ("Several Algorithms have been developed . . . Running times have ranged from O(mn) for the brute force algorithm to O(kn) or O(nlog(m)")

Pet. ('179) 47-48.

427. For the '441 Patent, Petitioner makes the same assertion:

> between successive notes of the melody."). Ghias further discloses that this search may be non-exhaustive: "it is considered desirable to use an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. *Id.* at 6:7-11, 6:23-35 ("Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from O(mn) for the brute force algorithm to O(kn) or O(nlog(m)"). This search

Pet. ('441) 47-48. Accordingly, the Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a non-exhaustive search for the same reasons set forth above in detail with respect the '179 Patent.

428. *Board concerns*: I note that the Board preliminary found that Ghias discloses the non-exhaustive search claim element for the '441 Patent for the same reason that the Board identified for '179 Patent:

> Ghias provides that "[t]he number of matches that the database 14 should retrieve depends upon the *error-tolerance* used during the key-search," and "the user can perform a *new query on a restricted search list consisting of songs just retrieved*. This allows the user to identify sets of songs that contain similar melodies." Ex. 1010, 6:63–65, 7:5–8 (emphases added). Thus, Ghias makes clear that the search need not be exhaustive, as Patent Owner has argued, and will act to "identify[] a close, but not necessarily exact or closest, match," per our claim construction.

Decision ('179) at 14.

> Ghias provides that "[t]he number of matches that the database 14 should retrieve depends upon the *error-tolerance* used during the key-search." Ex. 1010, 6:63–65 (emphasis added). Ghias further provides that "the user can perform a *new query on a restricted search list consisting of songs just retrieved*. This allows the user to identify sets of songs that contain similar melodies." *Id.* at 7:5–8 (emphasis added). Thus, Ghias makes clear that the search need not be exhaustive, as Patent Owner argues, and will act to "identify[] a close, but not necessarily exact or closest, match," per our claim construction.

Decision ('441) at 14. The Board's reliance on the "new query on a restricted search list" disclosed in Ghias to satisfy Petitioner's burden of demonstrating that the instituted clams are unpatentable based on Ghias fails for the same two reasons with respect to the '179 Patent:

Reason 1: Had these passages cited by the Board disclosed the claimed non-exhaustive search (they do not), it would be improper for the Board to rely on these passages to find '179 claims unpatentable because they were not identified in the Petition as support for the non-exhaustive search element.

Reason 2: Using a query on the "restricted search list consisting of songs just received" does not disclose the claimed non-exhaustive search.

2. neighbor search (claims 1, 13, 25).

429. I note that in instituting this Ground, the Board did not specifically address whether Ghias discloses the claimed neighbor search. Ghias does not

disclose a neighbor search. As explained above in detail with respect to the '179 Patent:

(a) a "neighbor search" is a search that identifies "a close, but not necessarily exact or closest, match," and

(b) Ghias does not disclose a neighbor search because the disclosed search always (necessarily) identifies an exact or the closest match.

430. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches "neighbor search." As support for this element, the Petition, Declaration, and corresponding charts rely on the same discussion and citations to Ghias identified for this claim element with respect to the '179 Patent addressed above. Compare Pet. ('179) 47-48 with Pet. ('441) 47-48; Moulin Decl. ('179) ¶122 with Moulin Decl. ('441) ¶122.

431. For the '179 Patent, Petitioner asserts:

> Ghias further discloses that this search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." *Id.* at 2:50-59, 6:60-63.

Pet. ('179) 47-48. For the '411 Patent, Petitioner similarly asserts:

> This search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." *Id.* at 2:50-59, 6:60-63

Pet. ('441) 47-48. Accordingly, the Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a neighbor search for the same reasons set forth above in detail with respect to the '179 Patent.

VIII. Signature.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Respectfully submitted,

Dated: September 14, 2015

Exhibit A

George Karypis *Professor*	Department of Computer Science & Engineering 4-192 EE/CS 200 Union Street SE Minneapolis, MN 55455	phone fax email URL	(612) 626-7524 (612) 626-1597 karypis@cs.umn.edu http://www.cs.umn.edu/~karypis

George Karypis' research interests span the areas of data mining, bioinformatics, cheminformatics, high performance computing, information retrieval, collaborative filtering, and scientific computing. His research has resulted in the development of software libraries for serial and parallel graph partitioning (METIS and ParMETIS), hypergraph partitioning (hMETIS), for parallel Cholesky factorization (PSPASES), for collaborative filtering-based recommendation algorithms (SUGGEST), clustering high dimensional datasets (CLUTO), finding frequent patterns in diverse datasets (PAFI), and for protein secondary structure prediction (YASSPP). He has coauthored over 250 papers on these topics and two books ("Introduction to *Protein Structure Prediction: Methods and Algorithms*" (Wiley, 2010) and "Introduction to Parallel Computing" (Publ. Addison Wesley, 2003, 2nd edition)). In addition, he is serving on the program committees of many conferences and workshops on these topics, and on the editorial boards of the IEEE Transactions on Big Data, ACM Transactions on Knowledge Discovery from Data, Data Mining and Knowledge Discovery, Social Network Analysis and Data Mining Journal, International Journal of Data Mining and Bioinformatics, the journal on Current Proteomics, Advances in Bioinformatics, and Biomedicine and Biotechnology.

PUBLICATIONS

Books

1. "Introduction to *Protein Structure Prediction: Methods and Algorithms*", Huzefa Rangwala and George Karypis (editors). Wiley Book Series on Bioinformatics, 2010.
2. "*Introduction to Parallel Computing*" (2nd edition). Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. Addison-Wesley, ISBN: 0-2016-4865-2, 2003.
3. "*Introduction to Parallel Computing: Design and Analysis of Algorithms*". Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. Benjamin/Cumming, ISBN: 0-8053-3170-0, 1994.

Book Chapters (Invited)

1. "*Mining Evolving Patterns in Dynamic Relational Networks*", Rezwan Ahmed and George Karypis, in Unsupervised Learning Algorithms (Editors Emre Celebi and Kemal Aydin). Springer, 2015 (in press).
2. "*Web Search-based on Ranking*". Andrea Tagarelli, Santosh Kabbur, and George Karypis, in Graph Analysis in Social Media (Editor Pitas Ioannis), CRC Press, 2015 (to appear).
3. "*A Comprehensive Survey of Neighborhood-Based Recommendation Methods*", Xia Ning, Christian Desrosiers, and George Karypis, in *Recommender Systems Handbook: 2nd edition* (Editors: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor). Springer, 2015 (to appear).
4. "*Big Data Frequent Pattern Mining*". David C. Anastasiu, Jeremy Iverson, Shaden Smith, and George Karypis, in Frequent Pattern Mining (Editor: Charu C. Aggarwal and Jawei Han). Springer, pp. 225—258, 2014.
5. "*Document Clustering: The Next Frontier*", David C. Anastasiu, Andrea Tagarelli, and George Karypis, in Data Clustering: Algorithms and Applications (Editor: Charu C. Aggarwal and Chandran K. Reddy), Chapman & Hall CRC, pp. 303—326, 2013.
6. "*METIS and ParMETIS*". George Karypis, Encyclopedia of Parallel Computing (Editor-in-Chief: David Padua), pp. 1117-1124, 2011.
7. "*A Comprehensive Survey of Neighborhood-based Recommendation Methods*". Christian Desrosiers and George Karypis, Recommender Systems Handbook, pp. 107-144, 2011.
8. "*Document Clustering*". Ying Zhao and George Karypis. In "Encyclopedia of Machine Learning", Claude Sammut (ed), Springer, 2010.
9. "*Scientific Data Analysis*", Chandrika Kamath, Nikil Wale, George Karypis, Gaurav Pandey, Vipin Kumar, Krishna Rajan, Nagiza F. Samatova, Paul Breimyer, Guruprasad Kora, Chongle Pan, and Srikanth Yoginath. In

"Scientific Data Management", Arie Shoshani and Doron Rotem (ed), CRC Press/Taylor and Francis Books, 2009.

10. *"Towards a Scalable kNN CF Algorithm: Exploring Effective Applications of Clustering"*. Al. Mamunur Rashid, Shyong K. Lam, Adam LaPitz, George Karypis, and John Riedl. In "Web Mining and Web Usage Analysis", O. Nasraoui, M. Spiliopoulou, J. Srivastava, B. Mobasher, and B. Masand, Springer, 2007.
11. *"Protein Structure Prediction Using String Kernels"*. Huzefa Rangwala, Kevin DeRonne, and George Karypis. In "Knowledge Discovery in Bioinformatics: Techniques, Methods, and Applications", Y. Pan and T. Hu (eds). John Wiley and Sons, 2007.
12. *"Data Mining Algorithms for Virtual Screening of Bioactive Compounds"*. Mukund Deshpande, Michihiro Kuramochi, and George Karypis. In "Data Mining in Biomedicine", P. Pardalos (eds). Springer-Verlag London Ltd, 2007.
13. *"Finding Topological Frequent Patterns from Graphs Datasets"*. Michihiro Kuramochi and George Karypis. In "Mining Graph Data", L.B. Holder and D. Cook (eds). John Wiley & Sons, 2006.
14. *"Criterion Functions for Clustering on High Dimensional Data"*. Ying Zhao and George Karypis. In "Grouping Multidimensional Data: Recent Advances in Clustering", Jacob Kogan Charles Nicholas, Marc Teboulle (eds). Springer-Verlag London Ltd, 2006.
15. *"Partitioning and Load Balancing For Emerging Parallel Applications and Architectures"*, Karen Devine, Erik G. Boman, and George Karypis. In "Parallel Processing for Scientific Computing", M. Heroux, P. Raghavan, and H. D. Simon (eds) SIAM, 2006.
16. *"Mining Scientific Datasets Using Graphs"*. Michihiro Kuramochi, Mukund Desphande, and George Karypis. In "Data Mining: Next Generation Challenges and Future Directions", H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), AAAI Press, 2004.
17. *"Mining Chemical Compounds"*. Mukund Deshpande, Michihiro Kuramochi, and George Karypis. In "Data Mining in Bioinformatics", J. Wang, M. Zaki, H. Toivonen, and D. Shasha (eds.). Springer-Verlag London Ltd, 2004.
18. *"Clustering in Life Sciences"*. Ying Zhao and George Karypis. In "Functional Genomics: Methods and Protocols", M. Brownstein, A. Khodursky and D. Conniffe (editors). Humana Press, 2003.
19. *"Multilevel Hypergraph Partitioning"*. George Karypis. In "Multilevel Optimization in VLSI CAD", J. Cong and J. R. Shinnerl (editors). Kluwer Academic Publishers, Boston, 2003.
20. *"Graph Partitioning For High Performance Scientific Simulations"*. Kirk Schloegel, George Karypis, and Vipin Kumar. In "Sourcebook of Parallel Computing", J. Dongarra, I. Foster, G. Fox, K. Kennedy, A. White, L. Torczon, and W. Gropp (eds.). Morgan Kaufmann, 2002.
21. *"Parallel Data Mining Algorithms"*. Mahesh Joshi, Eui-Hong Han, George Karypis, and Vipin Kumar. In "Sourcebook of Parallel Computing", J. Dongarra, I. Foster, G. Fox, K. Kennedy, A. White, L. Torczon, and W. Gropp (eds.). Morgan Kaufmann, 2002.
22. *"Data Mining for Turbulent Flows"*. Eui-Hong Han, George Karypis, and Vipin Kumar. In "Data Mining for Scientific and Engineering Applications", C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu (eds.). Kluwer Academic Publishers, 2001.
23. *"Parallel Association Rules"*. Mahesh Joshi, Eui-Hong Han, George Karypis, and Vipin Kumar. In "Large-scale Parallel and Distributed Data Mining", M. Zaki, C. Ho (eds.). Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI), vol. 1759, 2000, Springer-Verlag
24. *"Scalable Parallel Algorithms for Sparse Linear Systems"*, Anshul Gupta, George Karypis, and Vipin Kumar. In "Parallel Computing in Optimization". A. Migdalas, P. Pardalos, S. Story (eds.). Kluwer Academic Publishers; pp73—98, 1997.
25. *"Scalable Parallel Algorithms for Unstructured Problems"*. Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. A. Ferreira and J.D.P. Rolim (eds.), "Parallel Algorithms for Irregular Problems: State of the Art"; Kluwer Academic Publishers; pp. 99—113, 1995.

Journal Papers

1. *"Evaluation of Connected-Component Labeling Algorithms for Distributed-Memory Systems"*. Jeremy Iverson, Chandrika Kamath, and George Karypis. Parallel Computing, 44, pp. 53—68, May 2015.
2. *"Algorithms for Mining the Coevolving Relation Motifs in Dynamic Networks"*. Rezwan Ahmed and George Karypis. ACM Transactions on Knowledge Discovery from Data, 2015 (in press).
3. *"User-specific Feature-based Similarity Models for Top-n Recommendation of New Items"*. Asmaa Elbadrawy and George Karypis. ACM Transactions on Intelligent Systems, 2015 (in press).

4. "*Multi-Threaded Modularity Based Graph Clustering using the Multilevel Paradigm*". Dominique LaSalle and George Karypis, Journal of Parallel and Distributed Computing, Vol. 76, pp. 66—80, February 2015.
5. "*MPI for Big Data: New Tricks for an Old Dog*". Dominique LaSalle and George Karypis. Parallel Computing, 40(10), pp. 754—767, 2014.
6. "*Exploring the Transcriptome Space of a Recombinant BHK Cell Line Through Next Generation Sequencing*". Kathryn C. Johnson, Andrew Yongky, Nandita Vishwanathan, Nitya M. Jacob, Karthik P. Jayapal, Chetan T. Goudar, George Karypis, and Wei-Shou Hu. Biotechnology and Bioengineering, 111(4), pp. 770—781, 2013.
7. "*Pareto optimal pairwise sequence alignment*". Kevin W. DeRonne and George Karypis. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Mar-Apr; 10(2), 481—493, 2013.
8. "*Coarse- and Fine-grained Models for Proteins: Evaluation by Decoy Discrimination*". Christopher Kauffman and George Karypis. Proteins, May; 81(5): 754—773, 2013.
9. "*A Segment-based Approach to Clustering Multi-Topic Documents*". Andrea Tagarelli and George Karypis. Knowledge and Information Systems, Vol. 34, pp. 563—595, 2013.
10. "*A novel two*-box search paradigm for query disambiguation". David C. Anastasiu, Byron J. Gao, Xing Jiang, and George Karypis, Internet and Web Information Systems, Vol. 16, No. 1, pp. 1—29, 2013.
11. "*Multivariate Analysis of Cell Culture Bioprocess Data – Lactate Consumption as Process Indicator*". Huong Le, Santosh Kabbur, Luciano Pollastrini, Ziran Sun, Keri Mills, Kevin Johnson, George Karypis, and Wei-Shou Hu. Journal of Biotechnology, Vol. 162, pp. 210—223, 2012.
12. "*Function Genomics of Nectar Production in Brassicaceae*". R. Bender, P. Klinkenberg, Z. Jiang, B. Bauer, G. Karypis, N. Nguyen, M. Perera, B. Nikolau, and C. Carter. Flora, 2007(7), pp. 491-496, 2012.
13. "*Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks*". Ahmed Rezwan and George Karypis. Knowledge and Information Systems, Vol 33, No. 3, pp. 603—630, 2012.
14. "*Multi-view Learning via Probabilistic Latent Semantic Analysis*". Fuzhen Zhuang, George Karypis, Xia Ning, Qing He, and Zhongzhi Shi. Information Sciences, 199, 20—30, 2012.
15. "*Improved Machine Learning Models for Predicting Selective Compounds*". Xia Ning, Michael Walters, and George Karypis. Journal of Chemical Information and Modeling, 52 (1), pp. 38—50, 2012.
16. "*Computational Tools for Protein-DNA Interactions*". Chris Kauffman and George Karypis. WIREs Data Mining and Knowledge Discovery, 2: 14—28, 2012.
17. "*In Silico Structure-Activity-Relationship (SAR) Models From Machine Learning: A Review*". Xia Ning and George Karypis, Drug Development Research, Vol. 72, 2011.
18. "*Genome-wide Inference of Regulatory Networks in Streptomyces Coelicolor*". Marlene Castro-Melchor, Salim Charaniya, George Karypis, Eriko Takano, and Wei-Shou Hu. BMC Genomics, Vol. 11, pp. 578, 2010 (highly accessed).
19. "*Assessing Synthetic Accessibility of Chemical Compounds Using Machine Learning Methods*". Yevgeniy Podolyan, Michael Walters, and George Karypis. Journal of Chemical Information and Modeling, Vol. 50, pp. 979—991, 2010.
20. "*Mining Manufacturing Data for Discovery of High Productivity Process Characteristics*". Salim Charaniya, Huong Le, Huzefa Rangwala, Keri Mills, Kevin Johnson, George Karypis, and Wei-Shou Hu. Journal of Biotechnology, Vol. 147, pp. 186—197, 2010.
21. "*TOPTMH: Topology Predictor for Transmembrane alpha-Helices*". Rezwan Ahmed, Huzefa Rangwala, and George Karypis. Journal of Bioinformatics and Computational Biology, Vol 8, pp. 39—57, 2010.
22. "*svmPRAT: SVM-based Protein Residue Annotation Toolkit*". Huzefa Rangwala, Chris Kauffman, and George Karypis. BMC Bioinformatics, Vol 10, pp. 439, 2009.
23. "*Multi-Assay-based Structure-Activity-Relationship Models: Improving Structure-Activity Models by Incorporating Activity Information from Related Targets*". Xia Ning, Huzefa Rangwala, and George Karypis. Journal of Chemical Information and Modeling, 49(11), pp. 2444-2456, 2009.
24. "*LIBRUS: Combined Machine Learning and Homology Information for Sequence-based Ligand-Binding Residue Prediction*". Chris Kauffman and George Karypis. Bioinformatics, 25(23), pp. 3099-3107, 2009.
25. "*Target Fishing for Chemical Compounds using Target-Ligand Activity data and Ranking based Methods*". Nikil Wale and George Karypis. Journal of Chemical Information and Modeling, 49(10), pp. 2190-2201, 2009.
26. "*Improved estimation of structure predictor quality*". Kevin W. DeRonne and George Karypis. BMC Structural Biology, 9:41, 2009 (Highly Accessed).
27. "*Mining Transcriptome Data for Function-Trait Relationship of Hyper Productivity of Recombinant Antibody*". Salim Charaniya, George Karypis, and Wei-Shou Hu. Biotechnology and Bioengineering, 102(6), 2009.

28. *"Common Pharmacophore Identification Using Frequent Clique Detection Algorithm"*, Yevgeniy Podolyan and George Karypis. Journal of Chemical Information and Modeling, 49, 13—21, 2009.
29. *"CONTOUR: An Efficient Algorithm for Discovering Discriminating Subsequences"*, Jianyong Wang, Yuzhou Zhang, Lizhu Zhou, George Karypis, and Charu C. Aggarwal. Data Mining and Knowledge Discovery, Vol. 18, pp. 1—29, 2009.
30. *"Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach"*. Al Mamunur Rashid, George Karypis, and John Riedl. SIGKDD Explorations, 10(2), 90--100, 2008.
31. *"Mining Bioprocess Data: Challenges and Opportunities"*, Salim Charaniya, Wei-Shou Hu, and George Karypis. Trends in Biotechnology, 26(12), 690—699, 2008.
32. *"Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification"*, Nikil Wale, Ian A. Watson, and George Karypis. Knowledge and Information Systems (KAIS) Journal, Vol 14, No. 3, pp. 347—375, 2008.
33. *"Indirect Similarity based Methods for Effective Scaffold-Hopping in Chemical Compounds"*, Nikil Wale, Ian Watson, and George Karypis. Journal of Chemical Information and Modeling, 48 (4), 730—741, 2008.
34. *"fRMSDPred: Predicting local RMSD between structural fragments using sequence information"*, Huzefa Rangwala and George Karypis. PROTEINS: Structure, Function, and Bioinformatics, 72(3), 1005—1018, 2008.
35. *"Conserved GU-rich elements mediate mRNA decay by binding to CUG-binding protein 1"*, I.A. Vlasova , N.M. Tahoe, D. Fan, O. Larsson, B. Rattenbacher, J.R. SternJohn, J. Vasdewani, G. Karypis, C.S. Reilly, P. Bitterman, and P.R. Bohjanen. Molecular Cell, 29, 263—270, 2008.
36. *"Transcriptome Dynamics Based Operon Prediction and Verification in Streptomyces Coelicolor"*, Salim Charaniya, Sarika Mehra, Wei Lian, Karthik Jayapal, George Karypis, and Wei-Shou Hu. Nucleic Acids Research, Vol. 35, No. 21, pp. 7222—7236, June 2007.
37. *"Clustering methodologies for identifying country core competencies"*. R.N. Kostoff, J.A. del Rio, H. D. Cortes, C. Smith, A. Smith, C. Wagner, L. Leydesdorff, G. Karypis, G. Malpohl, and R. Tshiteya. Journal of Information Science, Vol. 33, No. 1, pp. 21—40, 2007.
38. *"Out-of-Core Coherent Closed Quasi-Clique Mining from Large Dense Graph Databases"*, Zhiping Zeng, Jianyong Wang, Lizhu Zhou, and George Karypis. ACM Transactions on Database Systems, Vol 32, Issue 2, 2007.
39. *"Effective Optimization Algorithms for Fragment-Assembly Based Protein Structure Prediction"*. Kevin DeRonne and George Karypis. Journal of Bioinformatics and Computational Biology, Vol. 5, No. 2, pp. 335—352, 2007.
40. *"Discovering Frequent Geometric Subgraphs"*. Michihiro Kuramochi and George Karypis. Information Systems Journal, 32, pp. 1101-1120, 2007.
41. *"QCRNA 1.0: A database of quantum calculations for RNA catalysis"*, T.J. Giese, B.A. Gregersen, Y. Liu, K. Nam, E. Mayaan, A. Moser, K. Range, O.N. Faza, C.S. Lopez, A.R. de Lera, G. Schaftenaar, X. Lopze, T.S. Lee, G. Karypis, D.M. York. Journal of Molecular Graphics & Modeling, Vol. 25, No. 4, pp. 423—433, 2006.
42. *"Building Multiclass Classifiers for Remote Homology Detection and Fold Recognition"*. Huzefa Rangwala and George Karypis. BMC Bioinformatics, 7:455, 2006.
43. *"On Mining Instance-Centric Classification Rules"*. Jianyong Wang and George Karypis. IEEE Transactions on Knowledge and Data Engineering, 18(11), pp. 1497—1511, 2006.
44. *"Genomic view of systemic autoimmunity in MRLlpr mice"*. J. Liu, G. Karypis, KL. Hippen, AL. Vegoe, P. Ruiz, GS. Gilkeson, and TW. Behrens. Genes and Immunity, 7, pp. 156—168, 2006.
45. *"YASSPP: Better Kernels and Coding Schemes Lead to Improvements in Protein Secondary Structure Prediction"*. George Karypis. PROTEINS: Structure, Function, and Bioinformatics, 64(3), pp. 575—586, 2006.
46. *"On Efficiently Summarizing Transactions for Clustering"*. Jianyong Wang and George Karypis. Knowledge and Information Systems, Vol. 9, No. 1, pp. 19—37, 2006.
47. *"Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization"*, Navaratnasothie Selvakkumaran, and George Karypis. IEEE Transactions on CAD, 25(3), pp. 504—517, 2006.
48. *"Power source roadmaps using bibliometrics and database tomography"*. R.N. Kostoff, R. Tshiteya, K.M. Pfeil, J.A. Humenik, and G. Karypis. Energy, 30, pp. 709—730, 2005.
49. *"The structure and infrastructure of Mexico's science and technology"*. R.N. Kostoff, J.A. del Rio, H. D. Cortes, C. Smith, A. Smith, C. Wagner, L. Leydesdorff, G. Karypis, G. Malpohl, and R. Tshiteya. Technological Forecasting and Social Change, 72, pp. 798—814, 2005.
50. *"Prediction of Contact Maps Using Support Vector Machines"*. Ying Zhao and George Karypis. International Journal on Artificial Intelligence Tools, Vol. 14, Issue 5, pp.849—865, 2005.

51. *"Profile Based Direct Kernels for Remote Homology Detection and Fold Recognition"*. Huzefa Rangwala and George Karypis. Bioinformatics, Vol. 31, No. 23, pp. 4239—4247, 2005.
52. *"Finding Frequent Patterns in a Large Sparse Graph"*. Michihiro Kuramochi and George Karypis. Data Mining and Knowledge Discovery, Vol. 11, No. 3, 243—271, 2005.
53. *"Frequent Substructure Based Approaches for Classifying Chemical Compounds"*. Mukund Deshpande, Michihiro Kuramochi, Nikhil Wale, and George Karypis. IEEE Transactions on Knowledge and Data Engineering, Vo. 17, No. 8, 1036—1050, 2005.
54. *"Gene Classification Using Expression Profiles: A Feasibility Study"*. Michihiro Kuramochi and George Karypis. International Journal on Artificial Intelligence Tools. Vol. 14, No. 4, pp. 641—660, 2005.
55. *"Data Clustering in Life Sciences"*. Ying Zhao and George Karypis. Molecular Biotechnology, 31(1), pp. 55—80, 2005.
56. *"Hierarchical Clustering Algorithms for Document Datasets"*. Ying Zhao and George Karypis. Data Mining and Knowledge Discovery, Vol. 10, No. 3, pp. 141—168, 2005.
57. *"Finding Frequent Patterns Using Length-Decreasing Support Constraints"*. Masakazu Seno and George Karypis. Data Mining and Knowledge Discovery, Vol. 10, No. 3, pp. 197—228, 2005.
58. *"Expression Levels for Many Genes in Human Peripheral Blood Cells are Highly Sensitive to ex Vivo Incubation"*. E. Baechler, F. Batliwalla, G. Karypis, P. Gaffney, K. Moser, W. Ortmann, K. Espe, S. Balasubramanian, K. Hughes, J. Chan, A. Begovich, S. Chang, P. Gregersen, and T. Behrens. Genes and Immunity, Vo. 5, No. 5, pp. 347—353, 2004.
59. *"A Boolean Algorithm for Reconstructing the Structure of Regulatory Networks"*. Sarika Mehra, Wei-Shou Hu, and George Karypis. Metabolic Engineering, Vol. 6, No. 4, pp. 326—339, 2004.
60. *"Macromolecule Mass Spectrometry: Citation Mining of User Documents"*. Ronald Kostoff, Clifford Bedford, Antonio del Rio, Hector Cortes, and George Karypis. Journal of the American Society for Mass Spectrometry, Vol. 15, No. 3, pp. 281—287, March 2004.
61. *"Parallel Formulations of Tree-Projection-Based Sequence Mining Algorithm"*. Valerie Guralnik and George Karypis. Parallel Computing, Vol. 30, pp. 443—472, 2004.
62. *"wCLUTO: A Web-Enabled Clustering Toolkit"*. Matthew Rasmussen, Mukund Deshpande, George Karypis, Jim Johnson, John Crow, and Ernest Retzel. Plant Physiology, October 2003, Vol. 133, pp. 510—516.
63. *"Item-based Top-N Recommendation Algorithms"*. Mukund Deshpande and George Karypis. ACM Transactions on Information Systems. Volume 22, Issue 1, pp. 143—177, January 2004.
64. *"An Efficient Algorithm for Discovering Frequent Subgraphs"*. Michihiro Kuramochi and George Karypis. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 9, 1038—1051, 2004.
65. *"Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering"*. Ying Zhao and George Karypis. Machine Learning, 55, pp. 311-331, 2004.
66. *"Interferon-Inducible Gene Expression Signature in Peripheral Blood Cells of Patients with Severe SLE"*. E. Baechler, F. Batliwalla, G. Karypis, P. Gaffney, W. Ortmann, K. Espe, K. Shark, W. Grande, K. Hughes, V. Kapur, P. Gregersen, and T. Behrens. Proceedings of the National Academies of Sciences (PNAS). Vol. 100, No. 5, pp. 2610—2615, March 2003.
67. *"Selective Markov Models for Predicting Web-Page Accesses"*. Mukund Deshpande and George Karypis. ACM Transactions on Internet Technology, Vol. 4, Issue 2, pp. 163—184, May 2004.
68. *"Predicting the Performance of Randomized Parallel Search: An Application to Robot Motion Planning"*. Daniel Challou, Maria Gini, Vipin Kumar, and George Karypis. Journal of Intelligent and Robotic Systems. Volume 38, Number 1, pp. 31—53, September 2003.
69. *"Parallel Static and Dynamic Multi-Constraint Graph Partitioning"*. Kirk Schloegel, George Karypis, and Vipin Kumar. Concurrency and Computation: Practice and Experience. Volume 14, Issue 3, pages 219—240, 2002.
70. *"Privacy Risks in Recommender Systems"*. Naren Ramakrishnan, Benjamin J. Keller, Batul J. Mirza, Ananth Y. Grama, and George Karypis. IEEE Internet Computing, 54—62, Vol 5, No. 6, 2001.
71. *"Wavefront Diffusion and LMSR: Algorithms for Dynamic Repartitioning of Adaptive Meshes"*. Kirk Schloegel, George Karypis, and Vipin Kumar. IEEE Transactions on Parallel and Distributed Systems. Vol. 12, No. 5, 451—466, May 2001.
72. *"Multilevel k-way Hypergraph Partitioning"*. George Karypis and Vipin Kumar. VLSI Design, Vol. 11, No. 3, pp. 285—300, 2000.
73. *"Scalable Parallel Data Mining for Association Rules"*. Sam Han, George Karypis, and Vipin Kumar. IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 3, pp337—352, 2000.

74. "*Chameleon: A hierarchical Clustering Algorithms Using Dynamic Modeling*". George Karypis, Eui-Hong Han, and Vipin Kumar. IEEE Computer, Special Issue on Data Analysis and Mining, Vol. 32, No. 8, pp68—75, August 1999.
75. "*Parallel Multilevel k-way Partition Scheme for Irregular Graphs*". George Karypis and Vipin Kumar. SIAM Review, Vol. 41, No. 2, pp. 278—300, 1999.
76. "*Document Categorization and Query Generation on the World Wide Web Using WebACE*". D. Boley, M. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. AI Review, Vol. 11, pp 365—391, December 1999.
77. "*Partitioning-Based Clustering for Web document Categorization*". Daniel Boley, Maria Gini, Robert Gross, Eui-Hong Han, Kyle Hastings, George Karypis, Vipin Kumar, Bamshad Mobasher, and Jerome Moore. Decision support Systems, Vol. 27, No. 3, pp. 329—341, 1999.
78. "*Multilevel Hypergraph Partitioning: Application in VLSI Domain*". George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. IEEE Transactions on VLSI Systems, Vol. 7, No. 1, pp. 69—79, 1999.
79. "*A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs*". George Karypis and Vipin Kumar. SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999.
80. "*A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering*". George Karypis and Vipin Kumar. Journal of Parallel and Distributed Computing, Vol. 48, pp. 71—85, 1998.
81. "*Multilevel k-way Partitioning Scheme for Irregular Graphs*". George Karypis and Vipin Kumar. Journal of Parallel and Distributed Computing, Vol. 48, pp. 96—129, 1998.
82. "*Multilevel Diffusion Schemes for Repartitioning of Adaptive Meshes*". Kirk Schloegel, George Karypis, and Vipin Kumar. Journal of Parallel and Distributed Computing, Vol. 47, pp. 109—124, 1997.
83. "*Highly Scalable Parallel Algorithms for Sparse Matrix Factorization*". Anshul Gupta, George Karypis, and Vipin Kumar. IEEE Transactions on Parallel and Distributed Systems, Vol. 8, No. 5, pp. 502—520, 1997.
84. "*Unstructured Tree Search on SIMD Parallel Computers*". George Karypis and Vipin Kumar. IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 10, pp. 1057—1072, October 1994.

Conference Papers

1. "*L2Knng: Fast Exact K-Nearest Neighbor Graph Construction with L2-Norm Pruning*". David C. Anastasiu and George Karypis, 24th ACM International Conference on Information and Knowledge Management (CIKM), Melbourne, Australia, 2015.
2. "*A Memory Management System Optimized for BDMPI's Memory and Execution Model*". Jeremy Iverson and George Karypis, EuroMPI, 2015.
3. "*Efficient Nested Dissection for Multicore Architectures*". Dominique LaSalle and George Karypis, EuroPar 2015.
4. "*SPLATT: Efficient and Parallel Sparse Tensor-Matrix Multiplication*", Shaden Smith, Niranjay Ravindran, Nicholas D. Sidiropoulos, and George Karypis, 29th IEEE International Parallel & Distributed Processing Symposium, 2015.
5. "*Factorized Bilinear Similarity for Cold-Start Item Recommendations*". Mohit Sharma, Jiayu Zhou, Junling Hu, and George Karypis, 2015 SIAM International Conference on Data Mining, 2015.
6. "*Collaborative Multi-Regression Models for Predicting Students' Performance*". Asmaa Elbadrawy, R. Scott Studham, and George Karypis, 5th International Learning Analytics & Knowledge Conference (LAK15), 2015.
7. "*Understanding Computer Usage Evolution*". David C. Anastasiu, Al M. Rashid, Andrea Tagarelli, and George Karypis, 31st IEEE International Conference on Data Engineering (ICDE), 2015.
8. "*Signaling Adverse Drug Reactions with Novel Feature-based Similarity Model*". Fan Yang, Xiaohui Yu, and George Karypis, IEEE Conf. on Bioinformatics and Biomedicine (BIBM), pp. 593—596, 2014.
9. "*Memory-Efficient Parallel Computation of Tensor and Matrix Products for Big Tensor Decomposition*". Niranjay Ravindran, Nicholas Sidiropoulos, Shaden Smith, and George Karypis. In 28th Asilomar Conference on Signals, 2014.
10. "*Opportunities for data-drive cloud-based mobile optimization*". W. Myott, Thao Nguyen, A. Chandra, G. Karypis, and J. Weissman. Intl. Conf. on Collaboration Technologies and Systems (CTS), pp. 483—487, 2014.
11. "*HOSLIM: Higher-Order Sparse Linear Method for Top-N Recommender Systems*". Evangelia Christakopoulou and George Karypis. 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 38—49, 2014.
12. "*L2AP: Fast Cosine Similarity Search with Prefix L-2 Norm Bounds*". David Anastasiu and George Karypis. 30th IEEE International Conference on Data Engineering (ICDE), pp. 784—795, 2014.

13. *"A Versatile Graph-based Approach to Package Recommendation"*. Roberto Interdonato, Salvatore Romeo, Andrea Tagarelli, and George Karypis. IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2013. [Best student paper award]
14. *"FISM: Factored Item Similarity Models for Top-N Recommender Systems"*. Santosh Kabbur, Xia Ning, and George Karypis. 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2013.
15. *"Multi-Threaded Graph Partitioning"*. Dominique LaSalle and George Karypis. 27th IEEE Intl. Parallel & Distributed Processing Symposium (IPDPS), 2013.
16. *"AREM: A Novel Associative Regression Model Based on EM Algorithm"*. Zhinghua Jiang and George Karypis. 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2013.
17. *"Sparse Linear Methods with Side Information for Top-N Recommendations"*. Xia Ning and George Karypis. 6th ACM Conference on Recommender Systems (RecSys), 2012.
18. *"Fast and Effective Lossy Compression Algorithms for Scientific Datasets"*. Jeremy Iverson, Chandrika Kamath, and George Karypis. Euro-Par 2012.
19. *"Topic Modeling for Segment-based Documents"*. Giovanni Ponti, Andrea Tagarelli, and George Karypis. 20th Italian Symposium on Advanced Database Systems, 2012.
20. *"Discerning key parameters influencing high productivity and quality through recognition of patterns in process data."* Huong Le, Marlene Castro-Melchor, Christian Hakemeyer, Christine Jung, Berthold Szperalski, George Karypis, and Wei-Shou Hu. BMC Proceedings, 5(Suppl 8), p91, 2011.
21. *"SLIM: Sparse Linear Methods for Top-N Recommender Systems"*. Xia Ning and George Karypis. 11th IEEE International Conference on Data Mining (ICDM), 497—506, 2011.
22. *"Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks"*. Rezwan Ahmed and George Karypis. 11th IEEE International Conference on Data Mining (ICDM), 1—10, 2011.
23. *"A Statistical model for topically Segmented Documents"*. Giovanni Ponti, Andrea Tagarelli, and George Karypis. 14th International Conference on Discovery Science, Finland, 247—261, 2011.
24. *"Improved Machine Learning Models for Predicting Selective Compounds"*. Xia Ning, Michael Walters, and George Karypis. ACM Conference on Bioinformatics, Computational Biology and Biomedicine. Chicago, August 2011.
25. *"Automatic Detection of Vaccine Adverse Reactions by Incorporating Historical Medical Conditions"*. Zhonghua Jiang and George Karypis. ACM Conference on Bioinformatics, Computational Biology and Biomedicine. Chicago, August 2011.
26. *"Content-Based Methods for Predicting Web-Site Demographic Attributes"*. Santosh Kabbur, Eui-Hong Han, and George Karypis. 10th IEEE International Conference on Data Mining (ICDM), 2010.
27. *"Multi-task Learning for Recommender Systems"*. Xia Ning and George Karypis. 2nd Asian Conference on Machine Learning (ACML), 2010.
28. *"A Novel Approach to Compute Similarities and its Application to Item Recommendation"*. Christian Desrosiers and George Karypis. 11th Pacific Rim International Conference on Artificial Intelligence (PRICAI), pp. 39—51, 2010 (Best paper award).
29. *"Within-network classification using local structure similarity"*. Christian Desrosiers and George Karypis. ECML-PKDD, pp. 260—275, 2009.
30. *"The Set Classification Problem and Solution Methods"*. Xia Ning and George Karypis. SIAM International Conference on Data Mining, 2009.
31. *"A Kernel Framework for Protein Residue Annotation"*. Huzefa Rangwala, Chris Kauffman, and George Karypis. Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery (PAKDD), pp. 439—451, 2009.
32. *"Genome Alignments using MPI-LAGAN"*. Ruinan Zhang, Huzefa Rangwala, and George Karypis. IEEE International Conference on Bioinformatics and Biomedicine, 2008.
33. *"TOPTMH: Topology Predictor for Transmembrane Alpha Helices"*. Rezwan Ahmed, Huzefa Rangwala, and George Karypis. ECML PKDD, pp. 23—38, 2008.
34. *"Improving Homology Models for Protein-Ligand Binding Sites"*. Chris Kauffman, Huzefa Rangwala, and George Karypis. LSS Computational systems Bioinformatics Conference (CSB), 2008.
35. *"Architecture Aware Partitioning Algorithms"*. Irene Moulitsas and George Karypis. 8th Intl. Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), pp. 42—52, 2008.
36. *"An Analysis of Information Content Present in Protein-DNA Interactions"*. Chris Kauffman and George Karypis. Pacific Symposium on Biocomputing, pp. 477—488, 2008.

37. *"fRMSDAlign: Protein Sequence Alignment Using Predicted Local Structure Information"*. Huzefa Rangwala and George Karypis. 6th Asia Pacific Bioinformatics Conference, pp. 111—122, 2008.
38. *"Interleaving of Gate Sizing and Constructive Placement for Predictable Performance"*. Sungjae Kim, Eugene Shragowitz, George Karypis, and Rung-Bin Lin. International Symposium on VLSI Design, Automation, and Test, pp. 1—4, 2007.
39. *"Methods for Effective Virtual Screening and Scaffold-Hopping in Chemical Compounds"*. Nikil Wale, Ian A. Watson, and George Karypis. LSS Computational Systems Bioinformatics Conference (CSB), pp. 403—416, 2007.
40. *"fRMSDPred: Predicting Local RMSD Between Structural Fragments using Sequence Information"*. Huzefa Rangwala and George Karypis. LSS Computational Systems Bioinformatics Conference (CSB), pp. 311—322, 2007.
41. *"Discriminating Subsequence Discovery for Sequence Clustering"*. Jianyong Wang, Yuzhou Zhang, Lizhu Zhou, George Karypis, and Charu C. Aggarwal. SIAM International Conference on Knowledge and Data Discovery, 2007.
42. *"Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification"*. Nikil Wale and George Karypis. IEEE International Conference on Data Mining (ICDM), pp. 678—689, 2006.
43. *"Incremental Window-based Protein Sequence Alignment Algorithms"*. Huzefa Rangwala and George Karypis. Bioinformatics Special Issue on the 5th European Conference on Computational Biology (ECCB), Vol 15, No. 4, e17—23, 2007.
44. *"Coherent Closed Quasi-Clique Discovery from Large Dense Graph Databases"*. Zhiping Zeng, Jianyong Wang, Lizhu Zhou, and George Karypis. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 797—802, 2006.
45. *"Effective Optimization Algorithms for Fragment-Assembly Based Protein Structure Prediction"*. Kevin W. DeRonne and George Karypis. LSS Computational Systems Bioinformatics Conference (CSB2006), 2006.
46. *"Multilevel Algorithms for Partitioning Power-Law Graphs"*. Amine Abou-Rjeili and George Karypis. IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2006.
47. *"Partitioning Algorithms for Parallel Applications on Heterogeneous Architectures"*. Irene Moulitsas and George Karypis. In the 2006 SIAM Conference on Parallel Processing for Scientific Computing, 2006.
48. *"Feature-based Recommendation System"*. Eui-Hong Han and George Karypis. Proceedings of the 14th Conference of Information and Knowledge Management (CIKM), pp. 446—452, 2005.
49. *"Partitioning Algorithms for Simultaneously Balancing Iterative and Direct Methods"*. Irene Moulitsas and George Karypis. In the 2005 SIAM Conference on Parallel Processing for Scientific Computing, 2005.
50. *"Effective Document Clustering for Large Heterogeneous Law Firm Collections"*. Jack G Conrad, Khalid Al-Kofahi, Ying Zhao, and George Karypis. 10th International Conference on Artificial Intelligence and Law (ICAIL), pp. 177—187, 2005.
51. *"HARMONY: Efficiently Mining the Best Rules for Classification"*. Jianyong Wang and George Karypis. Proceedings of the 2005 SIAM International conference on Data Mining, pp. 205—216, 2005.
52. *"Topic-Driven Clustering for Document Datasets"*. Ying Zhao and George Karypis. Proceedings of the 2005 SIAM International conference on Data Mining, pp. 358—369, 2005.
53. *"Influence in Ratings-Based Recommender Systems: An Algorithm-Independent Approach"*. Al Mamunur Rashid, George Karypis, and John Riedl. Proceedings of the 2005 SIAM International conference on Data Mining, 2005.
54. *"Soft Clustering Criterion Functions for Partitional Document Clustering"*. Ying Zhao and George Karypis. Proceedings of the 13th Conference of Information and Knowledge Management (CIKM), pp. 246—247, 2004.
55. *"SUMMARY: Efficiently Summarizing Transactions for Clustering"*. Jianyong Wang and George Karypis. Proceedings of the 4th IEEE Conference on Data Mining (ICDM), 2004.
56. *"GREW A Scalable Frequent Subgraph Discovery Algorithm"*. Michihiro Kuramochi and George. Proceedings of the 4th IEEE Conference on Data Mining (ICDM), pp.439—442, 2004.
57. *"Efficient Closed Pattern Mining in the Presence of Tough Block Constraints"*. Krishna Gade, Jianyong Wang, and George Karypis. Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 138—147, 2004.
58. *"Multi-Resource Aware Partitioning Algorithms for FPGAs with Heterogeneous Resources"*, Navaratnasothie Selvakkumaran, Abishek Ranjan, Salil Raje, George Karypis. Proceedings of the 41st Design and Automation Conference (DAC), pp. 741—746, 2004.

59. *"Finding Frequent Patterns in a Large Sparse Graph"*. Michihiro Kuramochi and George Karypis. Proceedings of the 2004 SIAM International Conference on Data Mining, 2004.
60. *"BAMBOO: Accelerating Closed Itemset mining by Deeply Pushing the Length Decreasing Support Constraint"*. Jianyong Wang and George Karypis. Proceedings of the 2004 SIAM International Conference on Data Mining, 2004.
61. *"Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds"*. Mukund Deshpande, Michihiro Kuramochi, and George Karypis. Proceedings of the 3rd IEEE Conference on Data Mining (ICDM), 2003.
62. *"Intelligent Meta-Search Engine for Knowledge Management"*. Eui-Hong Han, George Karypis, and Doug Mewhort. Proceedings of the 12th Conference of Information and Knowledge Management (CIKM), pp. 492—495, 2003.
63. *"Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization"*. Navaratnasothie Selvakumaran and George Karypis. IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2003.
64. *"Multi-Constraint Mesh Partitioning for Contact/Impact Computations"*. George Karypis. Proceedings of the 2003 ACM/IEEE Conference on Supercomputing 2003.
65. *"Prediction of Contact Maps Using Support Vector Machines"*. Ying Zhao and George Karypis. Proceedings of the 3rd IEEE International Conference on Bioinformatics and Bioengineering (BIBE), pp. 26—33, 2003.
66. *"Discovering Frequent Geometric Subgraphs"*. Michihiro Kuramochi and George Karypis. Proceedings of the 2nd IEEE Conference on Data Mining (ICDM), pp. 258—265, 2002.
67. *"SLPminer: An Algorithm for Finding Frequent Sequential Patterns Using a Length-Decreasing Support Constraint"*. Masakazu Seno and George Karypis. Proceedings of the 2nd IEEE Conference on Data Mining (ICDM), pp. 418-425, 2002.
68. *"A Polynomial Time Approximation Scheme for Rectilinear Steiner Minimum Tree Construction in the Presence of Obstacles"*, Jian. Liu, Ying. Zhao, Eugene Shragowitz, and George Karypis. In 9th International Conference on Electronics, Circuits and Systems, pp. 781—784, 2002.
69. *"Evaluation of Hierarchical Clustering Algorithms for Document Datasets"*. Ying Zhao and George Karypis. Proceedings of the 11th Conference of Information and Knowledge Management (CIKM), pp. 515-524, 2002.
70. *"Using Conjunction of Attribute Values for Classification"*. Mukund Deshpande and George Karypis. Proceedings of the 11th Conference of Information and Knowledge Management (CIKM), pp. 356—364, 2002.
71. *"Multi-objective Circuit Partitioning for Cutsize and Path-Based Delay Minimization"*. Cristinel Ababei, Navaratnasothie Selvakkumaran, Kia Bazargan, and George Karypis. IEEE/ACM International Conference on Computer Aided Design (ICCAD), pp. 181—185, 2002.
72. *"Evaluation of Techniques for Classifying Biological Sequences"*. Mukund Deshpande and George Karypis. Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery (PAKDD), 2002.
73. *"Expert Agreement and Content Based Reranking in a Meta Search Environment using Mearf"*. Uygar Oztekin, George Karypis, and Vipin Kumar. Proceedings of the 11th WWW Conference, pp. 333—344, 2002.
74. *"Incremental SVD-Based Algorithms for Highly Scalable Recommender Systems"*. Badrul Sarwar, George Karypis, Joe Konstan, and John Riedl. Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT), 2002.
75. *"Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering"*. Badrul Sarwar, George Karypis, Joe Konstan, and John Riedl. Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT), 2002.
76. *"Improve Precategorized Collection Retrieval by Using Supervised Term Weighting Schemes"*. Ying Zhao and George Karypis. International Conference on Information Technology Coding and Computing, pp. 16—21, April 2002.
77. *"Gene Classification Using Expression Profiles: A Feasibility Study"*. Michihiro Kuramochi and George Karypis. Proceedings of the 2nd IEEE International Conference on Bioinformatics and Bioengineering (BIBE), pp. 191-200, 2001.
78. *"Evaluation of Item-based Top-N Recommendation Algorithms"*. George Karypis. Proceedings of the 10th Conference of Information and Knowledge Management (CIKM), pp. 247—254, 2001.
79. *"Graph Partitioning for Dynamic, Adaptive and Multi-phase Scientific Simulations"*. Kirk Schloegel, George Karypis, and Vipin Kumar. IEEE International Conference on Cluster Computing, pp. 271—273, 2001.
80. *"A Scalable Algorithm for Clustering Sequential Data"*. Valerie Guralnik and George Karypis. Proceedings of the 1st IEEE Conference on Data Mining, pp. 179—186, 2001.

81. "*LPMiner: An Algorithm for Finding Frequent Itemsets Using Length Decreasing Support Constraints*". Masakazu Seno and George Karypis. Proceedings of the 1st IEEE Conference on Data Mining, pp. 505-512, 2001.
82. "*Frequent Subgraph Discovery*". Michihiro Kuramochi and George Karypis. Proceedings of the 1st IEEE Conference on Data Mining, pp. 313-320, 2001.
83. "*Multilevel Algorithms for Generating Coarse Grids in Multigrid Methods*". Irene Moulitsas and George Karypis. Proceedings on Supercomputing 2001.
84. "*Parallel Algorithms for Sequence Mining*". Valerie Guralnik, Nivea Garg, and George Karypis. Proceedings of Europar, pp. 310—320, 2001.
85. "*Selective Markov Models*". Mukund Deshpande and George Karypis. SIAM Conference on Data Mining, 2001.
86. "*Item-Based Collaborative Filtering Recommendation Algorithms*". Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. WWW10, pp. 285—295, 2001.
87. "*Text Categorization Using Weight adjusted k-Nearest Neighbor Classification*". Eui-Hong Han, George Karypis, and Vipin Kumar. Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 53—65, 2001.
88. "*Analysis of Recommendation Algorithms for E-Commerce*". Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 158—167, 2000.
89. "*Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval & Categorization*". George Karypis and Eui-Hong Han. Proceedings of the 9th International Conference on Information and Knowledge Management, pp. 12—19, 2000.
90. "*A Unified Algorithm for Load-balancing Adaptive Scientific Simulations*". Kirk Schloegel, George Karypis, and Vipin Kumar. Proceedings of the 2000 ACM/IEEE Conference on Supercomputing, 2000.
91. "*Centroid-Based Document Classification: Analysis & Experimental Results*". Eui-Hong Han and George Karypis. Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 424—431, 2000.
92. "*Memory Management Techniques for Gang Scheduling*". William Leinberger, George Karypis, and Vipin Kumar. Europar 2000.
93. "*Parallel Multilevel Algorithms for Multi-Constraint Graph Partitioning*". Kirk Schloegel, George Karypis, and Vipin Kumar. Europar , pp. 296—310, 2000, "*Distinguished Paper*" award.
94. "*Job Scheduling in the Presence of Multiple Resource Requirements*". William Leinberger, George Karypis, and Vipin Kumar. Proceedings of the 1999 ACM/IEE Conference on Supercomputing, 1999.
95. "*Multi-Capacity Bin Packing Algorithms with Applications to Job Scheduling under Multiple Constraints*". William Leinberger, George Karypis, and Vipin Kumar. Proceedings of the International Conference on Parallel Processing, pp. 404—412, 1999.
96. "*A New Algorithm for Multi-objective Graph Partitioning*". Kirk Schloegel, George Karypis, and Vipin Kumar. Proceedings of Europar, pp. 322-331, 1999.
97. "*Multilevel k-way Hypergraph Partitioning*". George Karypis and Vipin Kumar. Proceedings of the 36th Design Automation Conference, pp. 343—348, 1999.
98. "*PSPASES: An Efficient and Scalable Parallel Direct Solver*". Mahesh V. Joshi, George Karypis, Vipin Kumar, Anshul Gupta, and Fred Gustavson. Proceedings of 9th SIAM Conference on Parallel Processing and Scientific Computing, 1999.
99. "*Dynamic Repartitioning of Adaptively Refined Meshes*". Kirk Schloegel, George Karypis, and Vipin Kumar. Proceedings of 9th SIAM Conference on Parallel Processing and Scientific Computing, 1999.
100. "*Multilevel Algorithms for Multi-Constraint Graph Partitioning*". George Karypis and Vipin Kumar. Proceedings of 10th Supercomputing Conference, pp. 1—13, 1998.
101. "*Dynamic Repartitioning of Adaptively Refined Meshes*". Kirk Schloegel, George Karypis, and Vipin Kumar. Proceedings of 10th Supercomputing Conference, pp. 1—8, 1998.
102. "*A Performance Study of Diffusive vs. Remapped Load-Balancing Schemes*". Kirk Schloegel, George Karypis, Vipin Kumar, Rupak Biswas, and Leonid Oliker. Proceedings of the 11th Intl. Conference on Parallel and Distributed Computing Systems, 1998.
103. "*ScalParC: A new Efficient and Scalable Parallel Classification Algorithm for Mining Large Datasets*". Mahesh Joshi, George Karypis, and Vipin Kumar. Proceedings of the 12th Intl. Parallel Processing Symposium, pp. 573—579, 1998.

104."*A High Performance Two Dimensional Scalable Parallel Algorithm for Solving Sparse Triangular System*" Mahesh Joshi, Anshul Gupta, George Karypis, and Vipin Kumar. Proceedings of the 4th Intl. Conference on High Performance Computing, pp. 137—143, 1997.

105."*Scalable Parallel Data Mining for Association Rules*". Eui-Hong Han, George Karypis, and Vipin Kumar. Proceedings of the 1997 ACM-SIGMOD Intl. Conference on Management of Data, pp. 277—288, 1997.

106."*Parallel Threshold-based ILU Factorization*". George Karypis and Vipin Kumar. Proceedings of 9th Supercomputing Conference, pp. 1—24, 1997.

107."*Repartitioning of Adaptive Meshes: Experiments with Multilevel Diffusion*". Kirk Schloegel, George Karypis, and Vipin Kumar. Proceedings of the Third Intl. Euro-Par Conference, 1997.

108."*Design and Implementation of a Scalable Parallel Direct Solver for Sparse Symmetric Positive Definite Systems: Preliminary Results*". Anshul Gupta, Fred Gustavson, Mahesh Joshi, George Karypis, and Vipin Kumar. Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing, 1997.

109."*A Coarse-Grain Parallel Formulation of Multilevel k-way Graph Partitioning Algorithm*". George Karypis and Vipin Kumar. Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing, 1997.

110."*WebACE: A Web Agent for Document Categorization and Exploration*". J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher. Proceedings of the 2nd Intl. Conference on Autonomous Agents, pp. 408—415, 1997.

111."*Multilevel Hypergraph Partitioning: Application in VLSI Domain*". George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Proceedings of the 34th Design and Automation Conference, pp. 526—529, 1997.

112."*Parallel Multilevel k-way Graph Partitioning*". George Karypis and Vipin Kumar. Proceedings of 8th Supercomputing Conference, 1996.

113."*Architecture, Algorithms and Applications for Future Generation Supercomputers*". Vipin Kumar, Ahmed Sameh, Ananth Grama, and George Karypis. Proceedings of the 6th Symposium on the Frontiers of Massively Parallel Computing, pp. 346—354, 1996.

114."*Parallel Multilevel Graph Partitioning*". George Karypis and Vipin Kumar. Proceedings of the 10th Intl. Parallel Processing Symposium, pp. 314—319, 1996.

115."*Analysis of Multilevel Graph Partitioning*". George Karypis and Vipin Kumar. Proceedings of 7th Supercomputing Conference, 1995.

116."*Multilevel Graph Partitioning and Sparse Matrix* Ordering". George Karypis and Vipin Kumar. Proceedings of the 1995 Intl. Conference on Parallel Processing, 1995.

117."*A High Performance Sparse Cholesky Factorization Algorithm for Scalable Parallel Computers*". George Karypis and Vipin Kumar. Proceedings of the 5th Symposium on the Frontiers of Massively Parallel Computation, pp. 204—213, 1995.

118."*A Highly Parallel Interior Point Algorithm. Extended Abstract*". George Karypis, Anshul Gupta, and Vipin Kumar. Proceedings of the 7th SIAM Conference on Parallel Processing, 1995.

119."*A Parallel Formulation of Interior Point Algorithms*". George Karypis, Anshul Gupta, and Vipin Kumar. Proceedings of 6th Supercomputing Conference, pp. 1057—1072, 1994.

120."*Efficient Parallel Mappings of a Dynamic Programming Algorithm: A Summary of Results*". George Karypis and Vipin Kumar. Proceedings of the 7th Intl. Parallel Processing Symposium, pp. 563—568, 1993.

121."*Unstructured Tree Search on SIMD Parallel Computers: A Summary of Results*". George Karypis and Vipin Kumar. Proceedings of the 4th Supercomputing Conference, pp. 453—462, 1992.

Workshop Papers

1. "*Mining Coevolving Induced Relational Motifs in Dynamic Networks*". Rezwan Ahmed and George Karypis. Workshop on Dynamic Networks (SDM-Networks), SIAM Data mining Conference, 2015.
2. "*NLMF: NonLinear Matrix Factorization Methods for Top-N Recommender Systems*". Santosh Kabbur and George Karypis. 7th ICDM International Workshop on Domain Driven Data Mining (DDDM), 2014.
3. "*BDMPI: Conquering BigData with Small Clusters using MPI*". Dominique Lasalle and George Karypis. Intl. Workshop on Data-Intensive Scalable Computing Systems, Supercomputing 2013.
4. "*Enhancing Link-Based Similarity Through the Use of Non-Numerical Labels and Prior Information*". Christian Desrosiers and George Karypis. 8th Workshop on Mining and Learning with Graphs, 2010.
5. "*Within-network classification using local structure similarity*". Christian Desrosiers and George Karypis. 7th Workshop on Mining and Learning with Graphs, 2009.

6. *"The Set Classification Problem and Solution Methods"*. Xia Ning and George Karypis. ICDM Workshop on Foundations of Data Mining, 2008.
7. *"Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach"*. Al M Rashid, George Karypis, and John Riedl. SIGKDD Workshop on Web Mining and Web Usage Analysis (WEBKDD), 2008.
8. *"A Segment-based Approach to Clustering Multi-Topic Documents"*. Andrea Tagarelli and George Karypis. Text Mining Workshop, SIAM Data mining Conference, 2008.
9. *"A Multi-Level Parallel Implementation of a Program for Finding Frequent Patterns in a Large Sparse Graph"*. Steve Reinhardt and George Karypis. 12th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS), 2007.
10. *"ClustKNN: A Highly Scalable Hybrid Model- and Memory-Based CF Algorithm"*. Al Mamunur Rashid, Shyong K. Lam, George Karypis, and John Riedl. WebKDD 2006 Workshop.
11. *"Finding Functionally Related Genes by Local and Global Analysis of MEDLINE Abstracts"*. Sigve Nakken and Christopher Kauffman, and George Karypis. SIGIR04 Bio Workshop: Search and Discovery in Bioinformatics. 2004.
12. *"Perimeter-Degree: A priori metric for directly measuring and homogenizing interconnection complexity in multilevel placement"*. Navaratnasothie Selvakumaran, Phiroze Parakh, and George Karypis. IEEE Conference on System Level Interconnect Prediction (SLIP), pp. 53—59, 2003,
13. *"Mining Scientific Datasets Using Graphs"*. Michihiro Kuramochi, Mukund Deshpande, and George Karypis. NSF Workshop on Next Generation Data-mining, 2002.
14. *"Automated Approaches for Classifying Structures"*. Mukund Deshpande, Michihiro Kuramochi, and George Karypis. SIGKDD Workshop on Bioinformatics, BIOKDD 2002.
15. *"A Scalable Algorithms for Clustering Protein Sequences"*. Valerie Guralnik and George Karypis. Workshop on Bioinformatics, KDD 2001.
16. *"Efficient Algorithms for Creating Product Catalogs"*. Michael Steinbach, George Karypis, and Vipin Kumar. KDD-2000 Workshop on Web Mining, SIAM Data Mining Conference, 2001.
17. *"A Feature Weight Adjustment Algorithm for Document Classification"*. Shrikanth Shankar and George Karypis. KDD-2000 Workshop on Text Mining.
18. *"Application of Dimensionality Reduction in Recommender System – A Case Study"*. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. WebKDD-2000 Workshop.
19. *"A Comparison of Document Clustering Techniques"*. Michael Steinbach, George Karypis, and Vipin Kumar. KDD-2000 Workshop on Text Mining.
20. *"Load Balancing Across Near-Homogeneous Multi-Resource Servers"*. William Leinberger, George Karypis, Vipin Kumar, Rupak Biswas. In 9th Heterogeneous Computing Workshop, pp. 60—71, 2000.
21. *"Clustering Based on Association Rule Hypergraphs"*. Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.
22. *"Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering"*. J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher. Proceedings of the 7th Workshop on Information Technologies and Systems, 1997.
23. *"Experiences with A Parallel Formulation of An Interior Point Algorithm"*. George Karypis, Anshul Gupta, and Vipin Kumar. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Vol. 22, pp 163—180, 1995.

INVITED TALKS

1. *"Big Data Research: Methods, Systems, and Applications"*, Chinese University of Hong Kong, Hong Kong, December 2014.
2. *"Top-N Recommender Systems: Revisiting Item Neighborhood Methods"*, Wayne State University, Detroit, October 2014.
3. *"Top-N Recommender Systems: Revisiting Item Neighborhood Methods"*, Samsung Research, December 2013.
4. *"Multilevel Hypergraph Partitioning"*, Synopsys Inc., December 2013.
5. *"Top-N Recommender Systems: Revisiting Item Neighborhood Methods"*, International Summer School on Trends in Computing, Tarragona, Spain, July 2014.
6. *"Top-N Recommender Systems: Revisiting Item Neighborhood Methods"*, Samsung Research, December 2013.

7. *"Multilevel Hypergraph Partitioning"*, Synopsys Inc., December 2013.
8. *"Multi-topic Document Modeling"*, Modeling and Statistical Methods for the Regulatory Assessment of Tobacco Products. FDA, December 2013.
9. *"Partitioning & Clustering Big Graphs"*, Workshop on Big Data Analytics, Microsoft Research, Cambridge, UK, May 2013.
10. *"Top-N Recommender Systems: Revisiting Item Neighborhood Methods"*, Big Data School, UTS, Sydney, Australia, April 2013.
11. *"Chemical Genetics and Recommender Systems – Different Problems but Similar Solutions"*, Nanjing University, China, December 2012.
12. *"Chemical Genetics and Recommender Systems – Different Problems but Similar Solutions"*, Tsinghua University, China, December 2012.
13. *"Chemical Genetics and Recommender Systems – Different Problems but Similar Solutions"*, Rutgers, March 2012.
14. *"Data Mining Research"*, Army Research Laboratory, Aberdeen, MD, March 2012.
15. *"Chemical Genetics"*, Computer Science Department, University of Illinois, Urbana Champaign, May 2012.
16. *"Advancing Chemical Genetics: Mining the Target-Ligand Activity Matrix"*, IBM T.J. Watson, December 2009.
17. *"Advancing Chemical Genetics: Mining the Target-Ligand Activity Matrix"*, University of Texas, Austin, April 2009.
18. *"Algorithms for Graph and Hypergraph Partitioning and They Applications"*, Conference on Graph Theory and Its Applications, Coimbatore, India, December 2008.
19. *"Biclustering Methods meets Formal Concept Analysis"*, Concept Lattices and Their Applications, Olumouc, Czech Republic, October 2008.
20. *"Drug and Probe Discovery and its Mathematical Challenges"*, DOE/NSF Workshop on the Mathematics for Analysis of Petascale Data, June 2008.
21. *"Trends in Bioinformatics"*, Tech Tune-up, University of Minnesota, June 2008.
22. *"Accelerating Drug Discovery: Methods for Effective Virtual Screening and Scaffold Hopping"*, Colloquium, University of Huston, April 2008.
23. *"Indirect Similarity Measures in Cheminformatics"*, Eli-Lilly, December 2007.
24. *"Mining Large Graphs"*, DyDAn Workshop on Associating Semantics with Graphs, Rutgers, April 2007.
25. *"Data Mining for Bioprocess Optimization"*, Genentech Corporation, March 2007.
26. *"Sub-structure-Based Virtual Screening and Retrieval Algorithms in Drug Discovery"*, Agency for Science, Technology, and Research, Bioinformatics Institute, Singapore, April 2006.
27. *"Discovering Knowledge from Life Sciences Literature: Opportunities, Challenges, and Success Stories"*, Keynote speech at the "Workshop in Knowledge Discovery from Life Sciences Literature" at PAKDD, Singapore, April 2006.
28. *"Data-Mining Opportunities in Bioinformatics"*, SAS Data-Mining Conference, October 2003.
29. *"Genomic Grid: Distributed Resources, Data, and Services"*, Data Mining and Exploration Middleware for Distributed and Grid Computing, September 2004, Minnesota Supercomputing Institute, University of Minnesota.
30. *"Classifying Chemical Compounds"*, Eli-Lilly, August 2003.
31. *"Data-Mining and Bioinformatics"*, St. Cloud State University, January 2003.
32. *"Data-Mining and Bioinformatics"*, Minnesota IT Leadership Forum, October 2002.
33. *"Clustering Documents and its Applications"*, 7th Annual Text Summit, Thompson Publishing, September 2002 (keynote speech).
34. *"Frequent Subgraph Discovery: Mining Scientific and Relational Data Sets"*, IPAM workshop on Scientific Data Mining, UCLA, January 2002.
35. *"Multilevel Algorithms for Circuit Partitioning"*, IPAM workshop on Multilevel Methods for VLSI Design, UCLA, December 2001.
36. *"Selective Markov Models"*, Honeywell Laboratories, March 2001.
37. *"Concept Indexing: A Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval & Categorization"*, IMA Workshop on Text Mining, Minneapolis, April 2000.
38. *"Text Mining"*, Purdue, Computer Science Department, April 2000.
39. *"Data Mining in Genomics"*, Incyte Pharmaceuticals, Palo Alto, April 2000.
40. "Genomic Computing Issues and Mining Gene Expression Data", IEEE CS/IEEE EMBS, Minneapolis, November 1999.

41. *"Multi-Constraint and Multi-Objective Graph Partitioning"*. AHPCRC workshop on Graph Partitioning, Minneapolis, October 1999.
42. *"Chameleon: Clustering Using Dynamic Modeling"*. AHPCRC workshop on Scientific Data Mining, Minneapolis, September 1999.
43. *"Data Mining Research at AHPCRC"*. Center for Army Analysis, Washington, D.C., September 1999.
44. *"Clustering and Classification of High Dimensional Data-Sets"*. Lawrence Livermore National Lab, November 1998.
45. *"Multi-Constraint Graph Partitioning"*. Lawrence Livermore National Lab, October 1998.
46. *"Multi-label Classification of Statutes Documents"*. WEST Publishing Group, September 1998.
47. *"Multilevel Nested Dissection: Experiences with Parallel Formulations"*. SIAM Conference on Linear Algebra, October 1997.
48. *"Multilevel Repartitioning of Adaptive Meshes"*. Army HPC Research Center Workshop on Unstructured Mesh Generation and Partitioning, October 1997
49. *"Parallel and Adaptive Graph Partitioning"*. Lawrence Livermore National Lab, April 1997.
50. *"Graph Algorithms and Data Mining"*. Pataflops Algorithm Workshop, April 1997.
51. *"Parallel k-way Mesh Partitioning. Workshop on Parallel Unstructured Grid Computations"*. Argonne National Lab, September 1996.
52. *"Experiences with a Parallel Formulation of an Interior Point Algorithm"*. DIMACS Workshop on Parallel Processing of Discrete Optimization Problems, February 1995.
53. *"Multilevel Graph Partitioning Algorithms"*. Cray Research, September 1994.

TUTORIALS

1. *"Computational Methods for DNA and Protein Sequence Analysis"*. Genomics Signal Processing and Statistics, College Station, TX, 2006.
2. *"Parallel Partitioning Software for Static, Dynamic, and Multi-phase Computations"*. Supercomputing 2001, November 2001, Denver, CO.
3. *"Data mining for Genomics"*. 1st SIAM Conference on Data Mining, April 2001, Chicago, Il.
4. *"Using METIS and ParMETIS"*. Army HPC Research Center's Workshop on *"Graph Partitioning and Applications: Current and Future Directions"*, October 1999

RESEARCH GRANTS

1. *"BIGDATA: IA: DKA: Collaborative Research: Learning Data Analytics: Providing Actionable Insights to Increase College Student Success"*. NSF, $1,219,736, 9/1/2014—8/31/2018 (with Nikos Sidiropoulos and Thomas Brothen).
2. *"Methods for Learning Analytics"*. Digital Technology Initiative Seed Grant, UMN, $75,000, 9/1/2014—8/31/2015 (with Nikos Sidiropoulos).
3. *"Towards Predicting the Evolution of Computing Usage"*. Intel Corporation, $75,000, 9/1/2014—8/31/2015.
4. *"High-Performance Distributed Big Data Processing"*. Army Research Office, $297,168, 09/01/2014—03/01/2018.
5. *"PFI:AIR-TT: Automated Out-of-Core Execution of Parallel Message-Passing Applications"*, NSF, $200,000, 08/15/2014—01/31/2016 (with Andrew Morrow).
6. *"Profile- and Setting-Aware Top-N Recommendation Algorithms"*. Samsung Information Systems, $50,000, 03/15/2014—03/15/2015.
7. *"Towards Predicting the Evolution of Computing Usage"*, Intel Corporation, $75,000, 9/1/2013—8/31/2014.
8. *"BIGDATA: Mid-Scale: DA: Collaborative research: Big Tensor Mining: Theory, Scalable Algorithms and Applications"*. NSF $866,845, 12/01/2012—11/30/2016 (with Nikos Sidiropoulos (PI)).
9. *"Time Sensitive Efficient and Scalable Recommendation Methods"*. PayPal Inc., $45,000, 9/15/2012—9/14/2013.
10. *"CSR: Medium: Enriching Mobile User Experience Through The Cloud"*, NSF, $700,000, 8/13/2012—8/12/2015 (with Jon Weissman (PI) and Abhishek Chandra).
11. *"SI2-SSE: Software Infrastructure for Partitioning Sparse Graphs on Existing and Emerging Computer Architectures"*. NSF, $499,784, 09/15/2010—08/31/2014 (with M. Whalen).
12. *"Enabling Scientific Discovery in Exascale Simulations"*. DOE, $459,000, 09/01/2010—08/31/2013.

13. *"Computational Methods to Advance Chemical Genetics by Bridging Chemical and Biological Spaces"*, NSF $854,732, 09/01/2009—08/31/2014 (with M.A. Walters).
14. *"Functional Genomics of Nectar Production in Brassicaceae"*, NSF, $1,336,289, 9/1/2008—8/31/2013 (with Clay Carter).
15. *"Discerning Pivotal High Productivity Characteristics through Recognition of Patterns in Process Data"*, GenenTech, $108,750, 12/1/2007—12/1/2008 (with Wei-Shou Hu).
16. *"Effective & Efficient Whole Genome Alignment Algorithms"*, IBM Rochester, $35,000, 6/1/2006—6/1/2007.
17. *"Classification Algorithms for Chemical Compounds"*, NIH, $1,149,001, 9/30/2005—9/30/2009.
18. *"SEI: Virtual Screening Algorithms for Bioactive Compounds Based on Frequent Substructures"*, NSF, $405,498, 9/1/2004—8/31/2009.
19. *"ITR: Graph Partitioning Algorithms for Complex Problems & Applications"*, NSF, $122,000, 8/25/2003—8/24/2005.
20. *"Summer Bioinformatics Institute"*, NSF/NIH, $498,596, 01/01/03—12/31/05 (with V. Kumar, J. Carlis, L. Ellis, A. Grosberg, V. Kapur, A. Odlyzko, H. Othmer, W. Pan, R. Phillips, E. Retzel, K. Silverstein, D. Truhlar, N. Young).
21. *"CAREER: Scalable Algorithms for Knowledge Discovery in Scientific Data Sets"*, NSF, $320,900, February 2002—January 2008.
22. *"Scalable Algorithms for Scientific Computations"*, Army Research Office, $520,000, Fall 2001—Fall 2006 (as part of AHPCRC).
23. *"Pathogenesis and Therapy of Chronic Lung Rejection"*, National Institute of Health, $1,479,387, Fall 2001—Fall 2006 (with M. Hertz, R. King, V. Kapur, E. Retzel, H. Chen, and K. Savik).
24. *"Autoimmune Biomarkers Collaboratory"*, NIH, $1,525,454 Fall 2001—Fall 2006 (with T. Behrens).
25. *"Discovery of Changes from the Global Carbon Cycle and Climate System Using Data Mining"*, NASA, $525,091, Spring 2001- Spring 2004 (with V. Kumar, S. Shekhar, S. Klooster, C. Potter, and A. Torregrosa).
26. *"CISE Research Instrumentation: Cluster Computing for Knowledge Discovery in Diverse Data Sets"*, National Science Foundation, $121,618, February 2000—January 2003 (with M. Gini, J. Riedl, J. Konstan, S. Shekhar, J. Srivastava).
27. *"Parallelization of KIVA"*, Army Research Office, $240,000, August 2000—July 2003 (with S. Garrick and V. Kumar)
28. *"Scientific Data Mining"*, Department of Energy, $120,000, March 2000—February 2001 (with V. Kumar).
29. *"Dynamic Feature Extraction and Data Mining for Analysis of Turbulent Flows"*, National Science Foundation, $1,462,500, October 1999—September 2002 (with V. Kumar, V. Interrante, G. Candler, I. Marusic, Longmire, S. Garrick).
30. *"Multi-Constraint Multi-Objective Graph Partitioning"*, National Science Foundation, $386,544, September 1999—August 2002 (with V. Kumar).
31. *"Scalable Parallel Algorithms for Irregular & Adaptive Computations"*, Department of Energy (Level II ASCI Initiative), $578,000, October 1998 – September 2001; (with V. Kumar).
32. *"Scalable Parallel Algorithms for Solving Sparse Linear Systems"*, Army Research Office, $230,000, September 1998 – August 2001; (with V. Kumar).
33. *"Graph Partitioning for Dynamic, Adaptive and Multi-Phase Computations"*, SGI/Cray, $55,000; January 1998 – December 1999; (with V. Kumar).
34. *"Load Balancing on the Information Power Grid"*, NASA, $40,000; May 1998 – September 1998; (with V. Kumar)
35. *"Scalable Data Mining Algorithms"*, Army Research Office (ASSERT); $75,000; May 1997 – April 2000; (with V. Kumar).

SOFTWARE DEVELOPED

METIS Serial software package fqor partitioning unstructured graphs and for computing fill reducing matrix re-orderings. METIS is used extensively in numerous application areas including scientific computing, parallel and distributed processing, operations research, geographical information systems, molecular biology, and data mining.
URL: http://www.cs.umn.edu/~metis/metis.

hMETIS Serial software package for partitioning hypergraphs. HMETIS is based on the multilevel paradigm and is able to quickly compute very high quality partitions of very large and irregular

	hypergraphs. It is used extensively to partition hypergraphs corresponding to VLSI circuits, in data mining for clustering, and to optimize the storage of databases on disks. URL: http://www.cs.umn.edu/~metis/hmetis
PARMETIS	An MPI-based parallel library for partitioning unstructured and adaptively refined meshes and for computing fill-reducing matrix re-orderings. It is a highly parallel implementation of the serial METIS package; with additional functionality to accommodate needs for partitioning and load balancing that exist only on parallel computations. URL: http://www.cs.umn.edu/~metis/parmetis.
PSPASES	An MPI-based library that implements a parallel sparse Cholesky-based direct solver. It incorporates a highly parallel multi-frontal Cholesky algorithm, as well as highly parallel algorithms for computing fill reducing orderings, symbolic factorization, and forward and backward substitution. URL: http://www.cs.umn.edu/~mjoshi/pspases.
SUGGEST	A collaborative filtering based top-N recommendation engine. It uses an efficient item-based model that adapts to the sparsity of the data set that leads to real-time high quality recommendations. URL: http://www.cs.umn.edu/~karypis/suggest.
MGRIDGEN	A highly optimized serial and parallel library for obtaining a sequence of successive coarse grids that is well suited for geometric multigrid methods. The quality of the elements of the coarse grids is optimized using a multilevel framework. The parallel library is based on MPI and is portable to a wide-range of architectures. URL: http://www.cs.umn.edu/~moulitsa/software.html.
CLUTO	A software package for clustering low- and high-dimensional data sets. It treats data clustering as an optimization problem that tries to optimize a particular clustering criterion function. It provides a variety of clustering criterion functions and various partitional and agglomerative clustering algorithms. URL: http://www.cs.umn.edu/~cluto.
gCLUTO	A cross-platform graphical user interface tool on top of the CLUTO library that allows the users to interactively load, cluster, and visualize their datasets. One of its key features is the extensive cluster visualization capabilities that include, tree, matrix, and an OpenGL-based mountain-view of the clustering solution. URL: http://www.cs.umn.edu/~cluto/gcluto.
wCLUTO	wCLUTO is a web-enabled data clustering application that is designed for the clustering and data-analysis requirements of gene-expression analysis. wCLUTO is also built on top of the CLUTO clustering library. Users can upload their datasets, select from a number of clustering methods, perform the analysis on the server, and visualize the final results. URL: http://cluto.ccgb.umn.edu.
PAFI	A software package for discovering frequent patterns in diverse datasets. It contains three main frequent pattern discovery algorithms that can be used to find frequent itemset, sequences, and graph patterns in large databases. URL: http://www.cs.umn.edu/~pafi.
YASSPP	A web-server for predicting the secondary structure of proteins from primary sequence. It is based on a cascaded SVM-based machine learning model that combines custom-designed kernel functions with evolutionary information. URL: http://yasspp.cs.umn.edu
AFGEN	AFGen is a program that takes as input a set of chemical compounds and generates their vector-space representation based on the set of fragment-based descriptors they contain. This vector-based representation can be used for different tasks in cheminformatics including similarity search, virtual screening, and library design. URL: http://glaros.dtc.umn.edu/gkhome/afgen/overview
MONSTER	A web-based server that provides a set of services for annotating residues with functional and

structural properties from sequence information only. The structural and functional annotations that are currently provided are secondary structure, transmembrane helices, disorder regions, solvent accessible surface area, DNA binding residues, contact order, and protein blocks.
URL: http://bio.dtc.umn.edu/monster

BDMPI BDMPI is a message passing library and associated runtime system for developing out-of-core distributed computing applications for problems whose aggregate memory requirements exceed the amount of memory that is available on the underlying computing cluster. BDMPI is based on the Message Passing Interface (MPI) and provides a subset of MPI's API along with some extensions that are designed for BDMPI's memory and execution model.
URL: http://glaros.dtc.umn.edu/gkhome/bdmpi/overview

Nerstrand Nerstrand is a multi-threaded multilevel graph clustering tool for generating clusterings with high modularity. It supports both finding a specified number of clusters/communities as well as detecting the number of clusters/communities.
URL: http://www-users.cs.umn.edu/~lasalle/nerstrand

SLIM SLIM is a library that implements a set of top-N recommendation methods based on sparse linear models. These models are a generalization to the traditional item-based nearest neighbor collaborative filtering approaches implemented in SUGGEST, and use the historical information to learn a sparse similarity matrix by combining an L2 and L1 regularization approach.
URL: http://glaros.dtc.umn.edu/gkhome/slim/overview

L2AP L2AP is a program that provides high-performance implementations of several methods for finding all pairs of vectors whose cosine similarity is greater than a user-specified threshold. These vectors are often sparse and high-dimensional, e.g., document-term vectors, user-item ratings, etc. The methods that are implemented include approaches developed by our group that prune the search space using L2 norm bounds (L2AP and L2AP-approx) and various other state-of-the-art approaches such as AllPairs, MMJoin, and IdxJoin.
URL: http://glaros.dtc.umn.edu/gkhome/l2ap/overview

PROFESSIONAL ACTIVITIES

Editorships

1. Associate Editor, IEEE Transactions on Big Data; 2015—present.
2. Associate Editor, ACM Transactions on Knowledge Discovery from Data; 2013—present.
3. Action Editor, Data Mining and Knowledge Discovery, Springer; 2013—present.
4. Associate Editor, IEEE Transactions on Knowledge and Data Engineering; 2010—2014.
5. Editorial Board Member, Social Network Analysis and Data Mining Journal; 2010—present.
6. Editorial Board Member, Journal of Biomedicine and Biotechnology; 2008—present.
7. Editorial Board Member, Advances in Bioinformatics; 2007—present.
8. Editorial Advisory Board Member, Current Proteomics, 2007—present.
9. Editorial Board Member, International Journal of Data Mining and Bioinformatics; 2005—present.
10. Associate Editor, IEEE Transactions on Parallel and Distributed Systems; 2003—2007.
11. Guest editor of the special issue of the ACM Transactions on Knowledge Discovery from Data on "Bioinformatics"; 2007.
12. Guest editor of the special issue of IEEE Computing in Science & Engineering on *"Data Mining in Science"*; 2002.
13. Guest editor of the special issue of *Parallel Computing Journal* on *"Graph Partitioning and Parallel Computing"*; 1999.

Leadership Roles in Conferences

1. Program Committee Co-Chair of the International Conference on Data Science and Advanced Analytics (DSAA 2014), Shanghai, China, November 2014.

2. Program Vice Chair of the International Conference on Parallel Processing (ICPP 2014), Minneapolis, MN, September 2014.
3. Publicity co-Chair of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Tainan, Taiwan, May 2014.
4. Program Committee co-Chair of the ACM Recommender Systems Conference (RecSys'13), Hong Kong, China, 2013.
5. Program Committee co-Chair of the 13th International Conference on Data Mining (ICDM), Dallas, TX, December 2013.
6. Program Committee co-Chair of the International Conference on Advanced Data Mining and Applications, Nanjing, China, 2012.
7. Panel Chair of the 11th International Conference on Data Mining (ICDM), Vancouver, Canada, December 2011.
8. Chair for Bioinformatics and Computational Biology (BICoB), 2010, 2011.
9. Area chair for SIAM Data Mining Conference, Minneapolis, MN, 2007.
10. Area chair for ECML/PKDD Conference, 2006, 2011.
11. General Chair of the 6th IEEE Symposium on Bioinformatics and Bioengineering (BIBE), Washington, 2006.
12. Chair of the 5th IEEE Symposium on Bioinformatics and Bioengineering Conference (BIBE), Minneapolis, 2005.
13. Co-Chair of the 4th IEEE Bioinformatics and Bioengineering Conference (BIBE), Taiwan, 2004.
14. Vice Chair of the Program Committee for the 5th IEEE International Conference on Data Mining, New Orleans, Louisiana, November 2005.

Conference Organizing Committee Memberships

1. SIAM Conference on Computation Science and Engineering, March 2001, Reno, Nevada.

Workshop Organizer

1. Member of the organizing committee of the ECML/PKDD workshop on "Knowledge Discovery in Health Care and Medicine (KD-HCM)", Athens, Greece, September 2011.
2. Program chair for the 9th IEEE International workshop on High Performance Computational Biology, which occurred during the IPDPS 2010 conference, April 2010.
3. Member of the organizing committee of the 6th SIGKDD workshop on Data Mining in Bioinformatics, which occurred during the SIGKDD 2006 Conference, August 2006.
4. Member of the organizing committee of the 3rd International Workshop on Mining Graphs, Trees, and Sequences (MGTS), which occurred during the ECML/PKDD 2005 Conference, October 2005.
5. Member of the organizing committee of the PAKDD workshop on *"Text Mining"*, which occurred during the 6th Pacific Asia Conference on Knowledge Discovery and Data Mining, May 2002.
6. Member of the organizing committee of the SIAM workshop on *"Data mining for Genomics"*, which occurred during the 1st SIAM Conference on Data Mining, April 2001.
7. Member of the organizing committee of the Army HPC Research Center's Workshop on *"Graph Partitioning and Applications: Current and Future Directions"*, October 1999.
8. Organizer of a mini-symposium on *"High Performance Data Mining"* at the *"9th SIAM Conference on Parallel Processing for Scientific Computing"*, 1999.
9. Member of the organizing committee of the Army HPC Research Center Workshop on *"Unstructured Mesh Generation and Partitioning"*, 1998.

Conference Program Committee Memberships

1. International Conference on Bioinformatics and Computational Biology (BICoB): 2009-present.
2. International Conference on Machine Learning and Applications (ICMLA): 2008.
3. European Conference on Computational Biology (ECCB): 2008
4. International Conference on Database and Expert Systems (DEXA): 2008.
5. International Symposium on Bioinformatics Research and Applications (ISIBRA): 2008.
6. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD): 2007—present.
7. International Conference on Genome Informatics (GIW): 2007—present.
8. ECML/PKDD Conference: 2006—present.
9. IEEE International Conference on Bioinformatics and biomedicine (BIBM): 2007—present.
10. ACM SIGKDD Conference on Knowledge Discovery and Data Mining: 2004—present.

11. IEEE International Conference on Data Mining (ICDM): 2004—present
12. IEEE Symposium on Bioinformatics and Bioengineering (BIBE): 2004—present
13. SIAM Data Mining Conference: 2003—present.
14. Conference of the American Association of Artificial Intelligence (AAAI): 2006
15. ACM Conference on Information and Knowledge Management (CIKM): 2006—present.
16. International Conference on Database Systems for Advance Applications (DAFSAA): 2006—2007
17. International Parallel and Distributed Processing Symposium (IPDPS): 2004, 2006—present
18. International World-Wide-Web Conference (WWW): 2003
19. International Conference on High Performance Computing (HiPC): 2004.
20. International Conference on Parallel Processing (ICPP): 2003.
21. Supercomputing Conference: 2002, 2007.

Workshop Program Committee Memberships

1. Workshops held in conjunction with the SIGKDD conference:
    1. Large Scale Recommender Systems and the Netflix Prize Competition: 2008.
    2. Workshop on Link discovery: Issues, Approaches and Applications (LinkKDD): 2005—2006.
    3. Open Source Data Mining Workshop (OSDM): 2005.
    4. Multi-Relational Data Mining (MRLDM): 2005.
    5. Workshop on Knowledge Discovery in the Web (WebKDD): 2005—2006, 2008.
    6. Workshop on Data Mining in Bioinformatics (BIOKDD): 2002—2006.
2. Workshops held in conjunction with the ICDE conference:
    1. Workshop on Data Engineering Methods in Bioinformatics (DEBI): 2009.
3. Workshops held in conjunction with the ICDM conference:
    1. High Performance Data Mining Workshop: 2009.
    2. Workshop on Data Mining in Bioinformatics: 2004.
4. Workshops held in conjunction with the SIAM Data Mining conference:
    1. Bioinformatics Workshop: 2004.
    2. Workshop on Clustering High Dimensional Data Sets and its Applications: 2002—2003.
    3. Spatial Data Mining: 2006
5. Workshops held in conjunction with VLDB:
    1. Workshop on Data Mining and Bioinformatics: 2006.
6. Workshops held in conjunction with ECML/PKDD:
    1. Parallel Data Mining (PDM): 2006.
    2. Mining and Learning on Graphs (MLG): 2007—2008.
7. Workshops held in conjunction with IPDPS:
    1. Workshop on High-Performance Grid Computing: 2003—2006.
8. International Workshop on *"Biological Data Management"*, (BIDM): 2004—2005.
9. International workshop on Geographic and Biological Data Management (GBDM): 2004.
10. International workshop on Distributed Data Mining in Life Sciences (LifeDDM): 2005.

Reviewer

1. Served as the reviewer for over five hundred papers in various journals (including ACM Transactions on Computational Biology and Bioinformatics, ACM Transactions on Information Systems, ACM Transactions on Internet Technology, Statistical Analysis and Data Mining, Bioinformatics, BMC Bioinformatics, Proteins, Data Mining and Knowledge Discovery, Journal of Combinatorics, Machine Learning Journal, Data and Knowledge Engineering, Pattern Analysis and Applications, Pattern Recognition, Knowledge and Information Systems, Parallel Computing, SIAM Journal on Scientific Computing, Acta Informatica, International Journal of Computer Mathematics, IEEE Transactions on Computers, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Computer Aided Design, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, IEEE Concurrency, Journal of Experimental Algorithms, Image and Vision Computing, IEEE Signal Processing Letters, IEEE Journal of Selected Topics in Signal Processing, IEEE Communications Letters, IEEE Systems, Man and Cybernetics) and conferences for which I have served on their program committee.

2. Served as an external reviewer for proposals submitted to NSF, DOE, ARL, ARO, NASA, State of Louisiana, and Science Foundation of Ireland (SFI), on multiple NSF review panels and NIH study sections, and participated on a site visit for SFI and Hellenic Quality Assurance agency.

DEGREES UNDER MY SUPERVISION

Ph.D. **Current**

1. Jeremy Iverson (passed WPE)
2. Dominique Lasalle (passed WPE)
3. David Anastasiu (passed WPE)
4. Asmaa El Badrawy (passed WPE)
5. Evangelia Christakopoulou (passed WPE)
6. Sara Morsy
7. Shaden Smith
8. Agoritsa Polyzou
9. Mohit Sharma

Completed

1. Sam Han (Fall 1999, with V. Kumar, currently employed at Persistent Systems Ltd, US)
2. Kirk Schloegel (Fall 1999, with V. Kumar, currently employed at Smart Social Media, Inc.)
3. Valery Guralnik (2001, with J. Srivastava, currently employed at Honeywell)
4. William Leinberger (2001, with V. Kumar, currently employed at General Dynamics)
5. Mukund Deshpande (2003, with J. Srivastava, currently employed at Persistent Systems Ltd, India)
6. Navaratnasothie Selvakumaran (2005, currently employed at Frequency Inc)
7. Irene Moulitsas (2005 with Y. Saad, currently at Cranfield University, UK)
8. Michihiro Kuramochi (2005, currently employed at Google Inc.)
9. Ying Zhao (2005, with D. Du, currently at Tsinghua University, China)
10. Irina Makarevitch (2005) (Applied Plant Sciences)
11. Huzefa Rangwala (2008) (currently at George Mason University)
12. Nikil Wale (2008) (currently employed at Nodality Inc.)
13. Xia Ning (2012) (currently employed at Indiana University Purdue University Indianapolis)
14. Kevin DeRonne (2013) (currently employed at IPNav LLC)
15. Zhonghua Jiang (2013) (currently employed at Goldman Sachs)
16. Chris Kauffman (2013) (currently at George Mason University)
17. Rezwan Ahmed (2014) (currently at Boston Scientific)
18. Santosh Kabbur (2015) (currently at Amazon.com)

M.S. **Completed**

1. Sushrut Karanjkar (Spring 1998)
2. Dalvinder Malhotra (Winter 1998)
3. Kapil Surlekar (Spring 1999)
4. William Leinberger (Spring 1999)
5. Shrikanth Shankar (spring 2000)
6. Md. Al Hasan (Fall 2001)
7. Ekta Sirohi (Fall 2002)
8. Masakazu Seno (spring 2002)
9. Qing Zhang (Fall 2002)
10. Chang Liu (Fall 2002)
11. Sai Chen (Summer 2003)
12. Rezwan Ahmed (Spring 2003)
13. Nivea Garg (Fall 2003)
14. Krishna Gades (Spring 2004)

15. Eunah Cho (Spring 2004)
16. Jay Vasdewani (Spring 2004)
17. Mahbubur Rahim Khan (Fall 2004)
18. Aris Goulalas-Divanis (Spring 2005)
19. Brian Wallenfelt (Spring 2006)

HONORS

- Best Student Paper Award, ICTAI 2013.
- Best Paper Award, PRICAI 2010.
- 10-year Highest Impact Award, ICDM 2010.
- Distinguished Paper Award at EuroPar 2000.
- Honorable Mention (2nd Place) at KDDCup 2000 competition.
- First Prize Award at Mannheim SuParCup 95 (European Supercomputing Conference)
- Cray Research Fellow for 1995-96.
- Graduate School Fellow University of Minnesota for 1992-93.

EDUCATION

1992-1996	UNIVERSITY OF MINNESOTA, Minneapolis, MN Ph.D. in Computer Science, Spring 1996. GPA 4.0/4.0 Dissertation title: "*Graph Partitioning and Its Applications to Scientific Computing*" Dissertation advisor: Vipin Kumar
1988-1992	UNIVERSITY OF MINNESOTA, Minneapolis, MN BS in Computer Science, Spring 1992, Cum Laude, GPA 4.0/4.0

PROFESSIONAL EXPERIENCE

Fall 2009 to present	Computer Science Department, University of Minnesota **PROFESSOR**
Summer 2004 Spring 2009	Computer Science Department, University of Minnesota **ASSOCIATE PROFESSOR**
Fall 1999 to Spring 2004	Computer Science Department, University of Minnesota **ASSISTANT PROFESSOR**
Summer 1996 to Fall 1999	Computer Science Department, University of Minnesota **RESEARCH ASSOCIATE**

TEACHING EXPERIENCE

1. "*Research Methods*", CSCI 8001/8002.
2. "*Introduction to Algorithms & Data Structures*", CSCI 4041.
3. "*Introduction to Parallel Computing*", CSCI 5451
4. "*Introduction to Data Mining*", CSCI 8475.
5. "*Computational Techniques for Genomics*", CSCI 5481
6. "*Systems Analysis of Biological Processes*", CHEN 8754
7. "*Summer Institute – Army HPC Research Center*", Summers of 1997 & 1998.

US008010988B2

(12) **United States Patent**
Cox

(10) **Patent No.:** **US 8,010,988 B2**
(45) **Date of Patent:** **Aug. 30, 2011**

(54) **USING FEATURES EXTRACTED FROM AN AUDIO AND/OR VIDEO WORK TO OBTAIN INFORMATION ABOUT THE WORK**

(76) Inventor: **Ingemar J. Cox**, London (GB)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 797 days.

(21) Appl. No.: **11/445,928**

(22) Filed: **Jun. 2, 2006**

(65) **Prior Publication Data**

US 2007/0041667 A1 Feb. 22, 2007

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/950,972, filed on Sep. 13, 2001, now Pat. No. 7,058,223.

(60) Provisional application No. 60/232,618, filed on Sep. 14, 2000.

(51) **Int. Cl.**
H04N 7/173 (2011.01)

(52) **U.S. Cl.** **725/110**

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,919,479 A	11/1975	Moon et al.
4,230,990 A	10/1980	Lert, Jr. et al.
4,450,531 A	5/1984	Kenyon et al.
4,495,526 A	1/1985	Baranoff-Rossine
4,499,601 A	2/1985	Matthews
4,511,917 A	4/1985	Kohler et al.
4,547,804 A	10/1985	Greenberg
4,634,966 A	1/1987	Nakatani et al.
4,639,779 A	1/1987	Greenberg
4,677,455 A	6/1987	Okajima
4,677,466 A	6/1987	Lert, Jr. et al.
4,682,370 A	7/1987	Matthews
4,697,209 A	9/1987	Kiewit et al.
4,739,398 A	4/1988	Thomas et al.
4,776,017 A	10/1988	Fujimoto
4,805,020 A	2/1989	Greenberg
4,843,562 A	6/1989	Kenyon et al.
4,918,730 A	4/1990	Schulze
5,210,820 A	5/1993	Kenyon
5,283,819 A	2/1994	Glick et al.
5,437,050 A	7/1995	Lamb et al.
5,481,294 A	1/1996	Thomas et al.
5,581,658 A	12/1996	O'Hagan et al.
5,594,934 A	1/1997	Lu et al.

(Continued)

OTHER PUBLICATIONS

Peter N. Yianilos, Excluded Middle Vantage Point Forest for Nearest Neighbor Search, Aug. 1, 1999, pp. 1-12.*

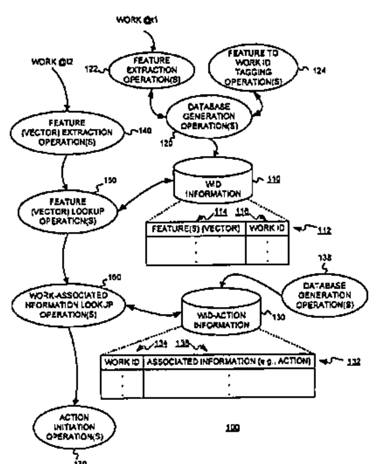
(Continued)

Primary Examiner — Brian T Pendleton
Assistant Examiner — Cai Chen
(74) *Attorney, Agent, or Firm* — Amster, Rothstein & Ebenstein LLP

(57) **ABSTRACT**

Information about an audio or video file played on a device is provided by (a) extracting features from the audio or video file, (b) communicating the features to a database, and (c) receiving the information about the audio or video file from the database. The information might include a song title, an album title, and/or a performer name. The information might include a title of a video work, a director of the video work, and/or names of performers in the video work. The information might be rendered on an output of the device. The information might be stored (e.g., persistently) locally on the device.

52 Claims, 10 Drawing Sheets



Google Ex. 1001

U.S. PATENT DOCUMENTS

5,629,739 A 5/1997 Dougherty
5,692,213 A 11/1997 Godlberg et al.
5,701,452 A 12/1997 Siefert
5,701,542 A 12/1997 Seifert
5,724,605 A 3/1998 Wissner
5,745,900 A 4/1998 Burrows
5,798,785 A * 8/1998 Hendricks et al. 725/46
5,850,490 A 12/1998 Johnson
5,918,223 A 6/1999 Blum et al.
5,953,415 A 9/1999 Nielsen
6,006,256 A 12/1999 Zdepski et al.
6,011,758 A 1/2000 Dockes et al.
6,026,439 A 2/2000 Chowdhury et al.
6,044,402 A 3/2000 Jacobson et al.
6,052,693 A 4/2000 Smith et al.
6,061,056 A 5/2000 Menard et al.
6,088,455 A 7/2000 Logan et al.
6,088,707 A 7/2000 Bates et al.
6,118,450 A 9/2000 Proehl et al.
6,119,124 A 9/2000 Broder et al.
6,169,986 B1 1/2001 Bowman
6,173,406 B1 1/2001 Wang
6,240,409 B1 5/2001 Aiken
6,243,725 B1 6/2001 Hempleman et al.
6,247,133 B1 6/2001 Palage et al.
6,253,193 B1 6/2001 Ginter et al.
6,263,348 B1 7/2001 Kathrow et al.
6,330,593 B1 12/2001 Roberts et al.
6,345,256 B1 2/2002 Milsted et al.
6,349,296 B1 2/2002 Broder et al.
6,360,215 B1 3/2002 Judd et al.
6,363,377 B1 3/2002 Kravets et al.
6,374,225 B1 4/2002 Hejna, Jr.
6,381,601 B1 4/2002 Fujiwara et al.
6,385,596 B1 5/2002 Wiser et al.
6,408,128 B1 * 6/2002 Abecassis 386/68
6,418,421 B1 7/2002 Hurtado et al.
6,449,226 B1 9/2002 Kumagai
6,452,874 B1 9/2002 Otsuka et al.
6,477,704 B1 11/2002 Cremia
6,496,802 B1 12/2002 Van Zoest et al.
6,505,160 B1 1/2003 Levy et al.
6,550,001 B1 4/2003 Corwin et al.
6,550,011 B1 4/2003 Sims, III
6,577,746 B1 6/2003 Evans et al.
6,591,245 B1 7/2003 Klug
6,598,228 B2 7/2003 Hejna, Jr.
6,609,105 B2 8/2003 Van Zoest et al.
6,654,757 B1 11/2003 Stern
6,665,661 B1 12/2003 Crow et al.
6,675,174 B1 1/2004 Bolle et al.
6,834,308 B1 * 12/2004 Ikezoye et al. 709/231
6,873,982 B1 3/2005 Bates et al.
6,931,451 B1 * 8/2005 Logan et al. 709/231
6,941,275 B1 9/2005 Swierczek
6,973,461 B1 12/2005 Fleming, III et al.
6,978,419 B1 12/2005 Kantrowitz
6,990,453 B2 1/2006 Wang et al.
7,013,301 B2 3/2006 Holm et al.
7,058,223 B2 * 6/2006 Cox 382/190
7,106,904 B2 9/2006 Shima
7,155,449 B2 12/2006 Pingel et al.
7,158,929 B2 1/2007 Wouters et al.
7,168,083 B2 1/2007 Kalker et al.
7,302,574 B2 11/2007 Conwell et al.
7,366,718 B1 4/2008 Pugh et al.
7,421,723 B2 9/2008 Harkness et al.
7,477,739 B2 1/2009 Haitsma et al.
7,523,312 B2 4/2009 Kalker et al.
7,587,728 B2 9/2009 Wheeler et al.
7,647,604 B2 1/2010 Ramaswamy
7,650,616 B2 1/2010 Lee
7,757,248 B2 7/2010 Harkness et al.
2001/0001160 A1 * 5/2001 Shoff et al. 725/51
2001/0003818 A1 6/2001 Pingel et al.
2002/0023020 A1 2/2002 Kenyon et al.
2002/0032698 A1 3/2002 Cox
2002/0120925 A1 * 8/2002 Logan 725/9
2002/0156760 A1 10/2002 Lawrence et al.
2003/0106017 A1 6/2003 Kanchirayappa et al.
2003/0146940 A1 * 8/2003 Ellis et al. 345/811
2004/0199387 A1 * 10/2004 Wang et al. 704/243
2005/0160363 A1 7/2005 Bhogal et al.
2006/0101069 A1 5/2006 Bell et al.
2006/0206462 A1 9/2006 Barber
2007/0041667 A1 2/2007 Cox
2007/0083510 A1 4/2007 McArdle
2007/0118375 A1 5/2007 Kenyon et al.
2008/0091684 A1 4/2008 Ellis et al.
2008/0250241 A1 10/2008 Ginter et al.

OTHER PUBLICATIONS

Baum, L., et al., "A Maximation Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *The Annals of Mathematical Statistics*, vol. 41, No. 1, pp. 164-171 (1970).

Dempster, A. P., et al., "Maximum Likelihood from Incomplete Data via the SEMS Algorithm", *Journal of the Royal Statistical Society*, Series B (Methodological), vol. 39, Issue 1, pp. 1-38 (1977).

Reynolds, D., et al., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, vol. 3, No. 1, pp. 72-83 (Jan. 1995).

Nievergelt, J., et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure." *ACM Transactions on Database Systems*, vol. 9, No. 1, pp. 38-71 (Mar. 1984).

Heintze, N., "Scalable Document Fingerprinting," *Proc. USENIX Workshop on Electronic Commerce* (1996).

Wold, E., et al., "Content-Based Classification, Search, and Retrieval of Audio," *IEEE Multimedia*, vol. 3, Issue 3, pp. 27-63 (1996).

Bhanu, B., et al., "Learning Feature Relevance and Similarity Metrics in Image Databases", *Proceedings of the IEEE Workshop on Content—Based Access of Image and Video Libraries*, pp. 14-19 (1998).

Del Bimbo, A., et al , "Using Weighted Spatial Relationships in Retrieval by Visual Contents", *Image Description and Retrieval*, pp. 161-192 (1998).

Indyk, P., and Motwani, R., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," *Proceeding of the Thirtieth Annual ACM Symposium on Theory of Computing*, pp. 604-613 (1998).

La Cascia, M., et al , "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web", *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 24-29 (1998).

Yianilos, N. P., "Excluded Middle Vantage Point Forests for Nearest Neighbor Search," *DIMACS Implementation Challenge*, ALENEX'99 (1999).

Yoshitaka, A., et al., "A Survey on Content-Based Retrieval for Multimedia Databases", *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, No. 1, pp. 81-93 (Jan./Feb. 1999).

Lawrence, S., et al., "Digital Libraries and Automonous Citation Indexing," *IEEE Computer*, pp. 67-71 (Jun. 1999).

Yianilos, N. P., "Locally Lifting the Curse of Dimensionality for Nearest Neighbor Search," *Symposium on Discrete Algorithms, Proceeding of the Eleventh Annual ACM-SIAM symposium on Discrete Algorithms*, pp. 361-370 (2000).

Kimura, A., et al., "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," *IEEE Conf. on Acoustics, Speech and Signal Processing*, (ICASSP '01), vol. 3, pp. 1429-1432 (2001).

Chavez, E., et al., "Searching in Metric Spaces", (Sep. 2001) *ACM Computing Surveys*, vol. 33, No. 3, pp. 273-321 (Sep. 2001).

Haitsma, J., et al., "Robust Audio Hashing for Content Identification, Int." *Workshop on Content Based Multimedia Indexing*, Brescia, Italy (Sep. 19-21, 2001).

Haitsma, J., and Walker, T., "A Highly Robust Audio Fingerprinting System," *Journal of New Music Research*, 1744-5027, vol. 32, Issue 2, pp. 211-221 (2003).

Schleimer, Saul, et al., "Winnowing: Local Algorithms for Document Fingerprinting ACM SIGMOD" (Jun. 9-12, 2003).

"Searching Near-Replicas of Images via Clustering" Edward Chang, Chen Li, James Wang, Peter Mork, Gio Wiederhold Proc. SPIE Symposium of Voice, Video, and Data Communications, 1999.

"Rime: A Replicated Image Detector for the World-Wide Web" Edward Y. Chang, James Ze Wang, Chen Li, and Gio Wiederhold, SPIE 1998.
"Safeguarding and charging for information on the internet," H. Garcia-Molina, S. Ketchpel, and N. Shivakumar, Proceedings of ICDE, 1998.
"Copy detection mechanisms for digital documents," S. Brin and H. Garcia-Molina, Proceedings of ACM SIG-MOD, May 1995.
"The x-tree: An index structure for high-dimensional data," S. Berchtold, Proceedings of the 22nd VLDB, Aug. 1996.
"The sr-tree: An index structure for high-dimensional nearest neighbor queries," N. Katayama and S. Satoh, Proceedings of ACM SIGMOD, May 1997.
"The k-d-b-tree: A search structure for large multidimensional dynamic indexes," J. T. Robinson, Proceedings of ACM SIGMOD, Apr. 1981.
"Query by image and video content: the QBIC system," M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and et al, IEEE Computer 28(9), pp. 23{32, 1995.
"Visual information retrieval," A. Gupta and R. Jain, Communications of the ACM 40(5), pp. 69-79, 1997.
"Visualseek: A fully automated content-based image query system," J. R. Smith and S.-F. Chang, ACM Multimedia Conference, 1996.
"Similarity indexing: Algorithms and performance," D. A. White and R. Jain, Proc. SPIE vol. 2670, San Diego, 1996.
"The r*-tree: an efficient and robust access method for points and rectangles," N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, Proceedings of ACM Sigmod, May 1990.
"R-trees: a dynamic index structure for spatial searching," A. Guttman, Proceedings of ACM Sigmod, Jun. 1984.
"Similarity indexing with the ss-tree," D. A. White and R. Jain, Proceedings of the 12th ICDE, Feb. 1996.
"The tv-tree: an index structure for high-dimensional data," K.-L. Lin, H. V. Jagadish, and C. Faloutsos, VLDB Journal 3 (4), 1994.
"M-tree: An efficient access method for similarity search in metric spaces," P. Ciaccia, M. Patella, and P. Zezula, Proceedings of the 23rd VLDB, Aug. 1997.
"Nearest neighbor queries," N. Roussopoulos, S. Kelley, and F. Vincent, Proceedings of ACM Sigmod, May 1995.
"An extensible hashing index for high-dimensional similarity search," C. Li, E. Chang, and J. Z. Wang, Stanford Technical Report, Aug. 1998. [NOT AVAILABLE].
"Two algorithms for nearest-neighbor search in high dimensions" J. M. Kleinberg, Proc 29th STOC, 1997.
"A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996.
"Adaptive Color Image Embeddings for Database Navigation" Yossi Rubner, Carlo Tomasi and Leonidas J. Guibas, Proceedings of the 1998 IEEE Asian Conference on Computer Vision.
A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces R. Weber, H-J Schek, S. Blott Proc., 24th VLDB Conf. 1998.

* cited by examiner

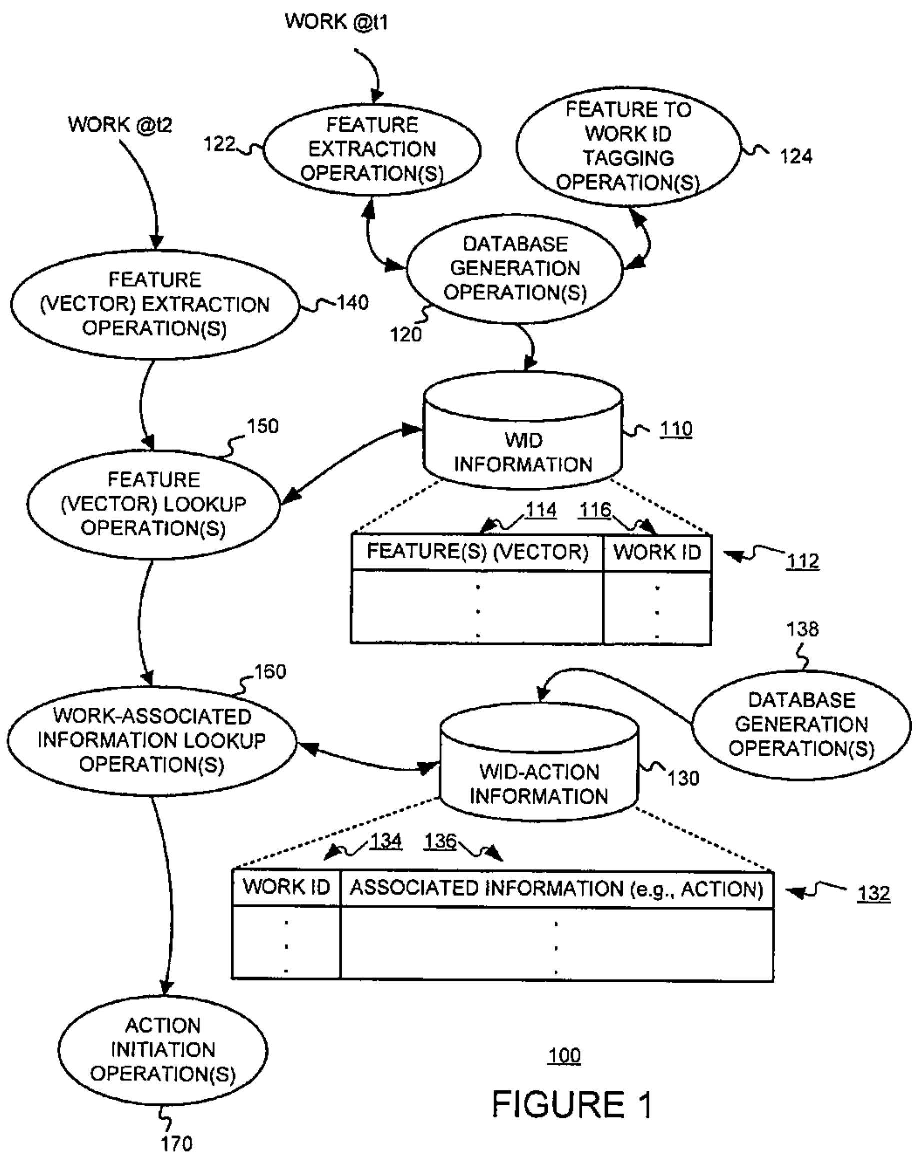

WORK @t1
WORK @t2
122
FEATURE EXTRACTION OPERATION(S)
FEATURE TO WORK ID TAGGING OPERATION(S)
124
DATABASE GENERATION OPERATION(S)
120
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140
150
FEATURE (VECTOR) LOOKUP OPERATION(S)
WID INFORMATION
110
114
116
FEATURE(S) (VECTOR)
WORK ID
112
138
DATABASE GENERATION OPERATION(S)
160
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WID-ACTION INFORMATION
130
134
136
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
132
ACTION INITIATION OPERATION(S)
170
100

FIGURE 1

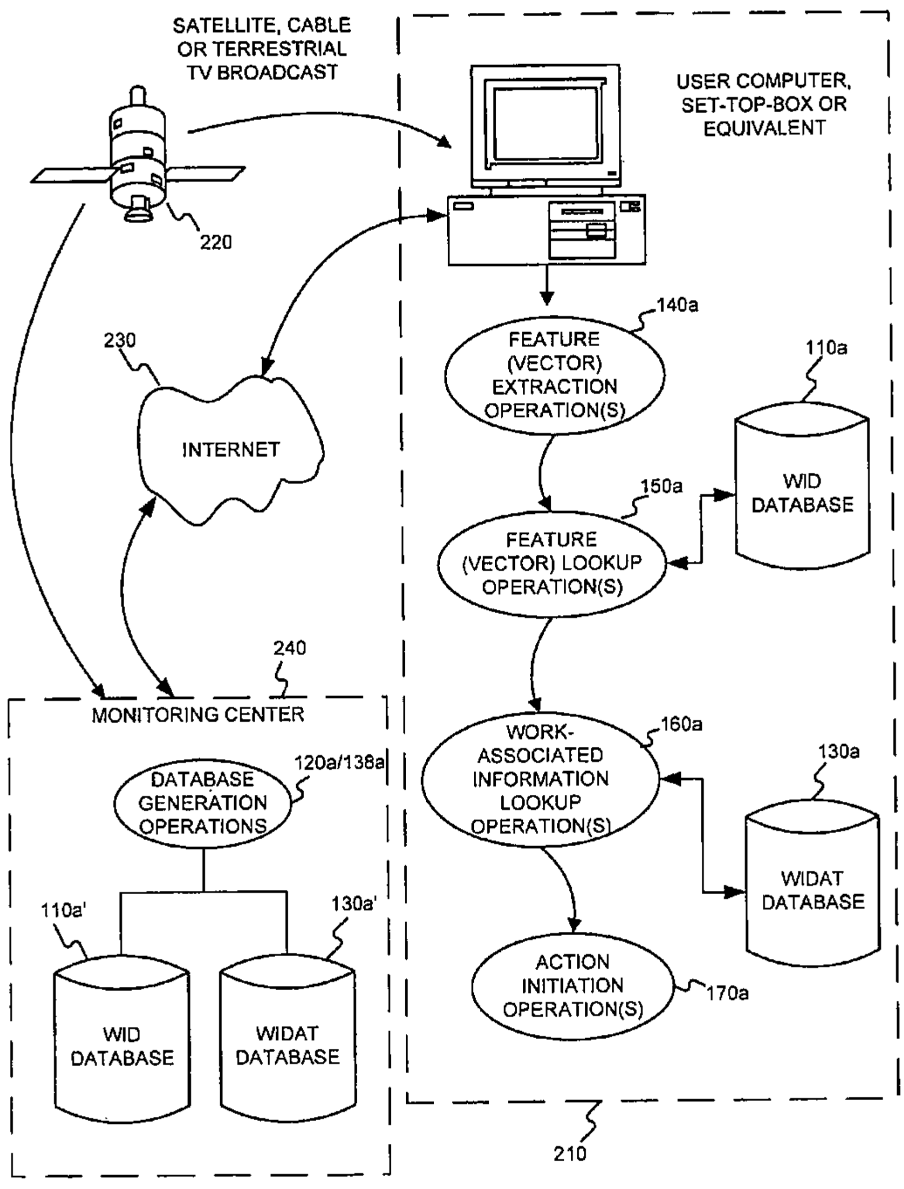

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
220
230
INTERNET
140a
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
110a
WID
DATABASE
150a
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
240
MONITORING CENTER
DATABASE
GENERATION
OPERATIONS
120a/138a
160a
WORK-
ASSOCIATED
INFORMATION
LOOKUP
OPERATION(S)
130a
WIDAT
DATABASE
110a'
130a'
WID
DATABASE
WIDAT
DATABASE
ACTION
INITIATION
OPERATION(S)
170a
210

FIGURE 2

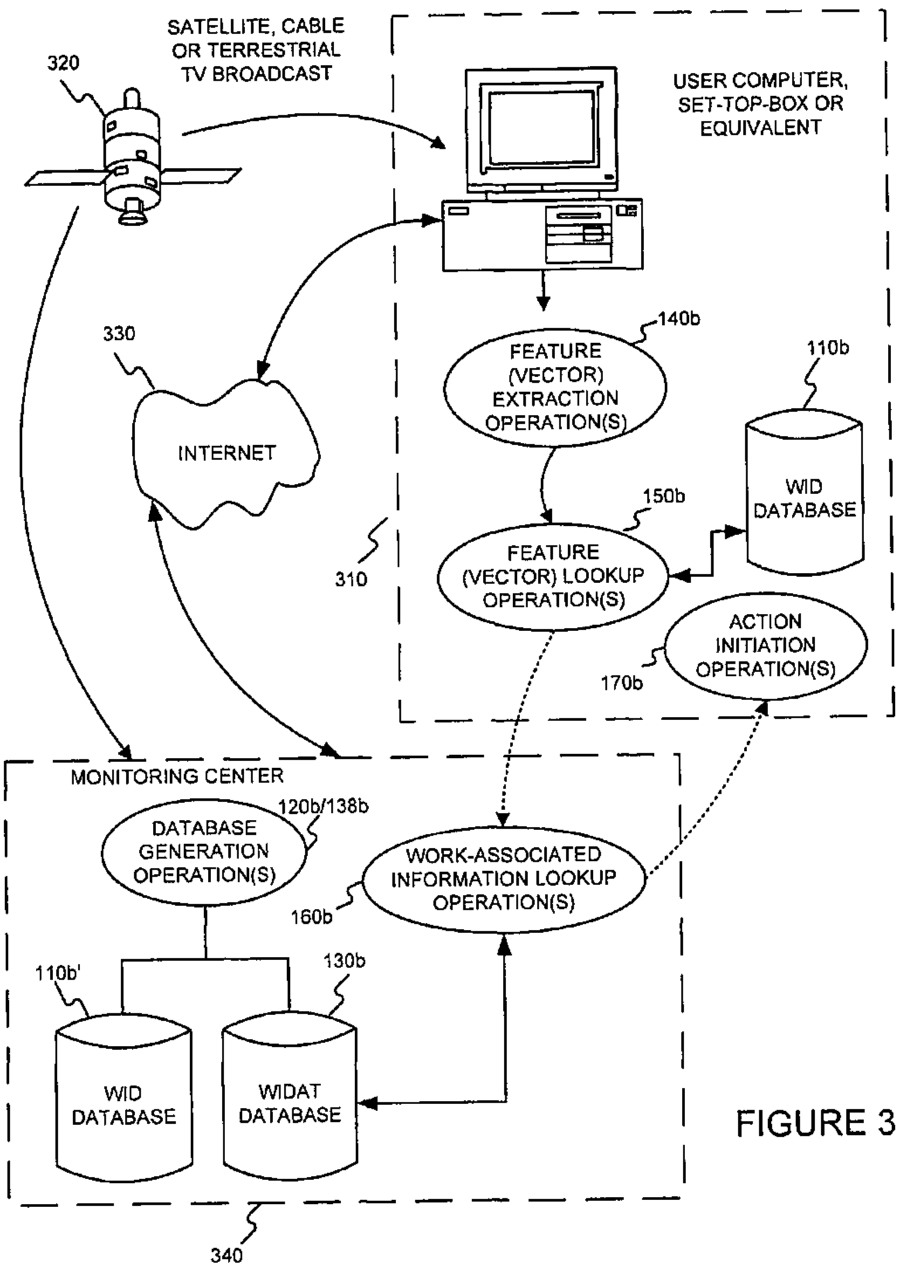

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
320
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
330
140b
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
110b
INTERNET
150b
WID
DATABASE
310
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
ACTION
INITIATION
OPERATION(S)
170b
MONITORING CENTER
120b/138b
DATABASE
GENERATION
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
160b
110b'
130b
WID
DATABASE
WIDAT
DATABASE
340

FIGURE 3

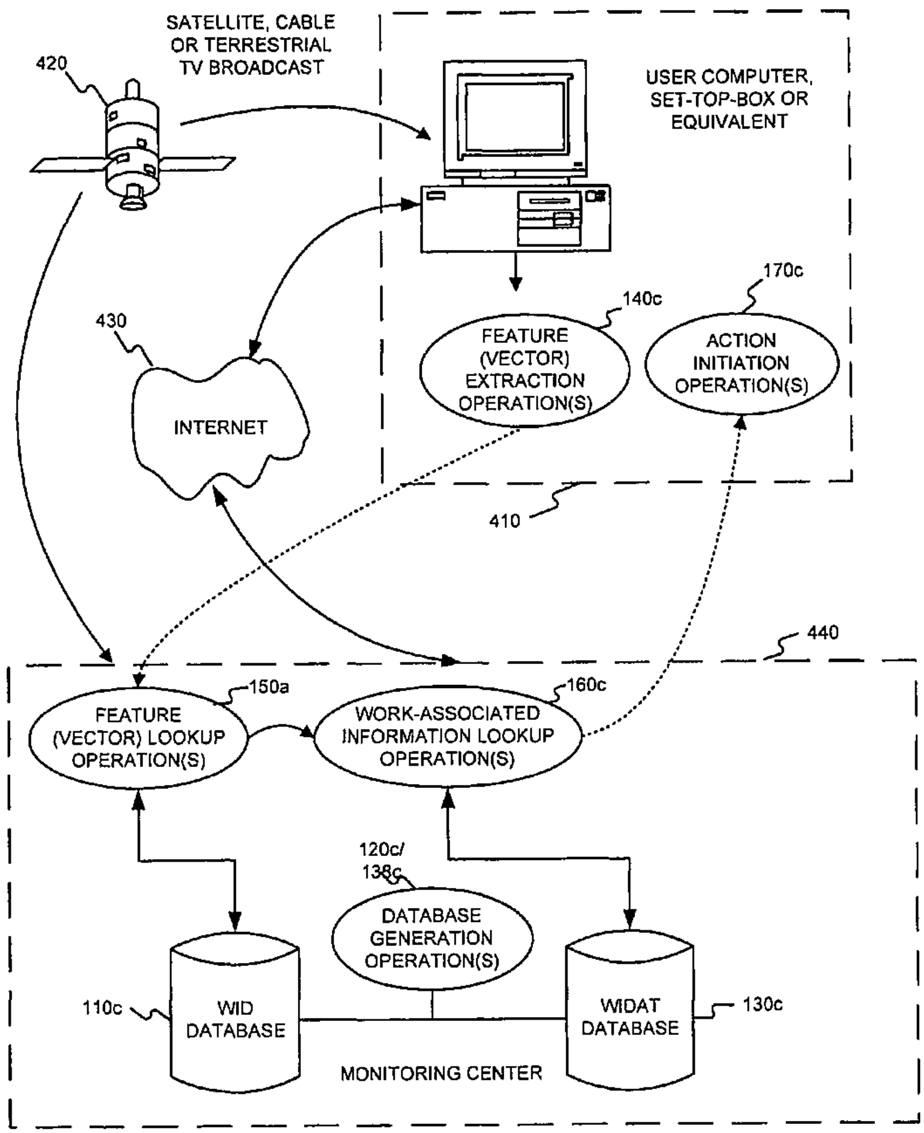

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
420
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
170c
140c
430
FEATURE (VECTOR) EXTRACTION OPERATION(S)
ACTION INITIATION OPERATION(S)
INTERNET
410
440
150a
160c
FEATURE (VECTOR) LOOKUP OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
120c/ 138c
DATABASE GENERATION OPERATION(S)
110c
WID DATABASE
WIDAT DATABASE
130c
MONITORING CENTER

FIGURE 4

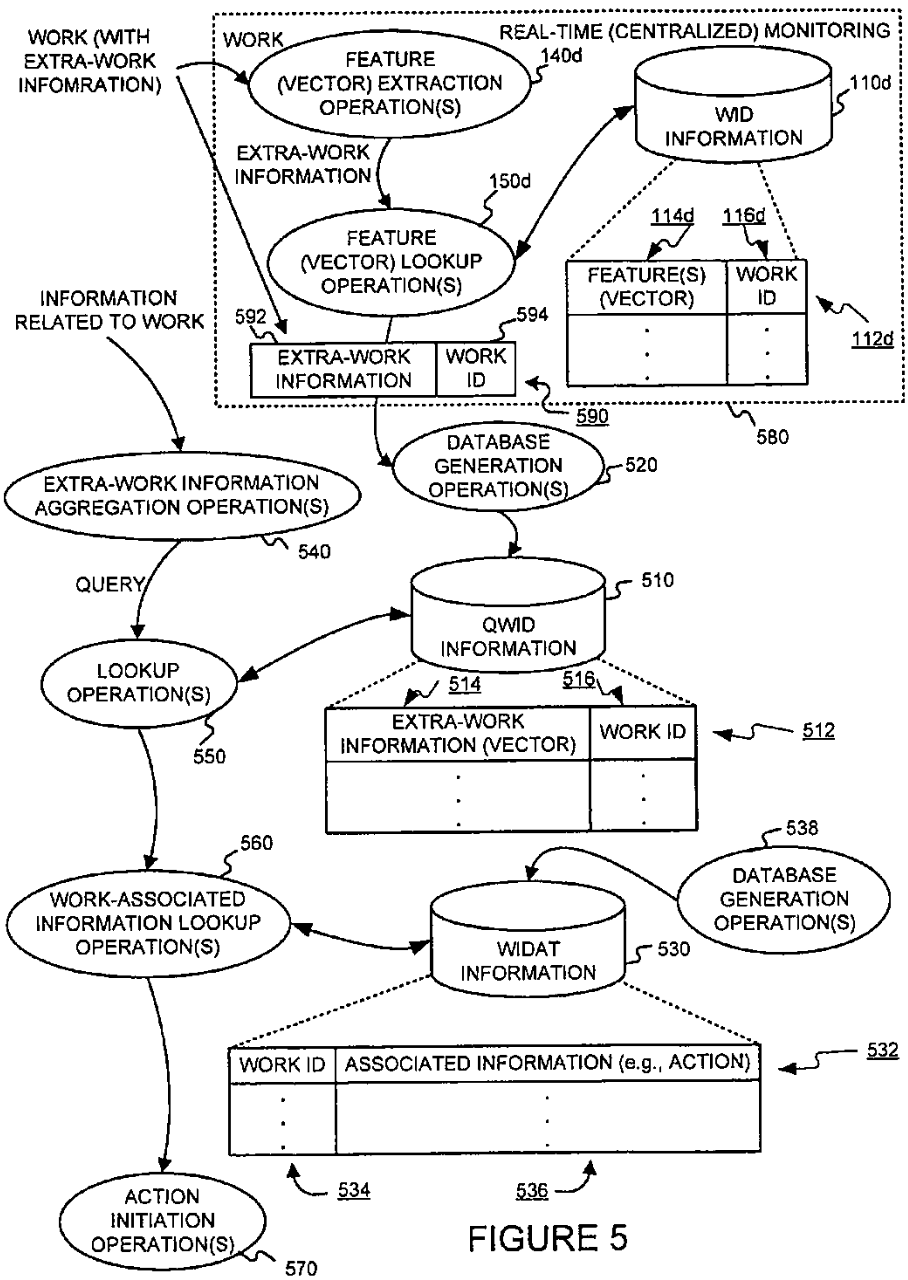

WORK (WITH EXTRA-WORK INFOMRATION)
WORK
REAL-TIME (CENTRALIZED) MONITORING
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140d
WID INFORMATION
110d
EXTRA-WORK INFORMATION
150d
FEATURE (VECTOR) LOOKUP OPERATION(S)
114d
116d
FEATURE(S) (VECTOR)
WORK ID
112d
INFORMATION RELATED TO WORK
592
594
EXTRA-WORK INFORMATION
WORK ID
590
580
DATABASE GENERATION OPERATION(S)
520
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
540
QUERY
510
QWID INFORMATION
LOOKUP OPERATION(S)
514
516
EXTRA-WORK INFORMATION (VECTOR)
WORK ID
512
550
538
560
DATABASE GENERATION OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WIDAT INFORMATION
530
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
532
534
536
ACTION INITIATION OPERATION(S)
570

FIGURE 5

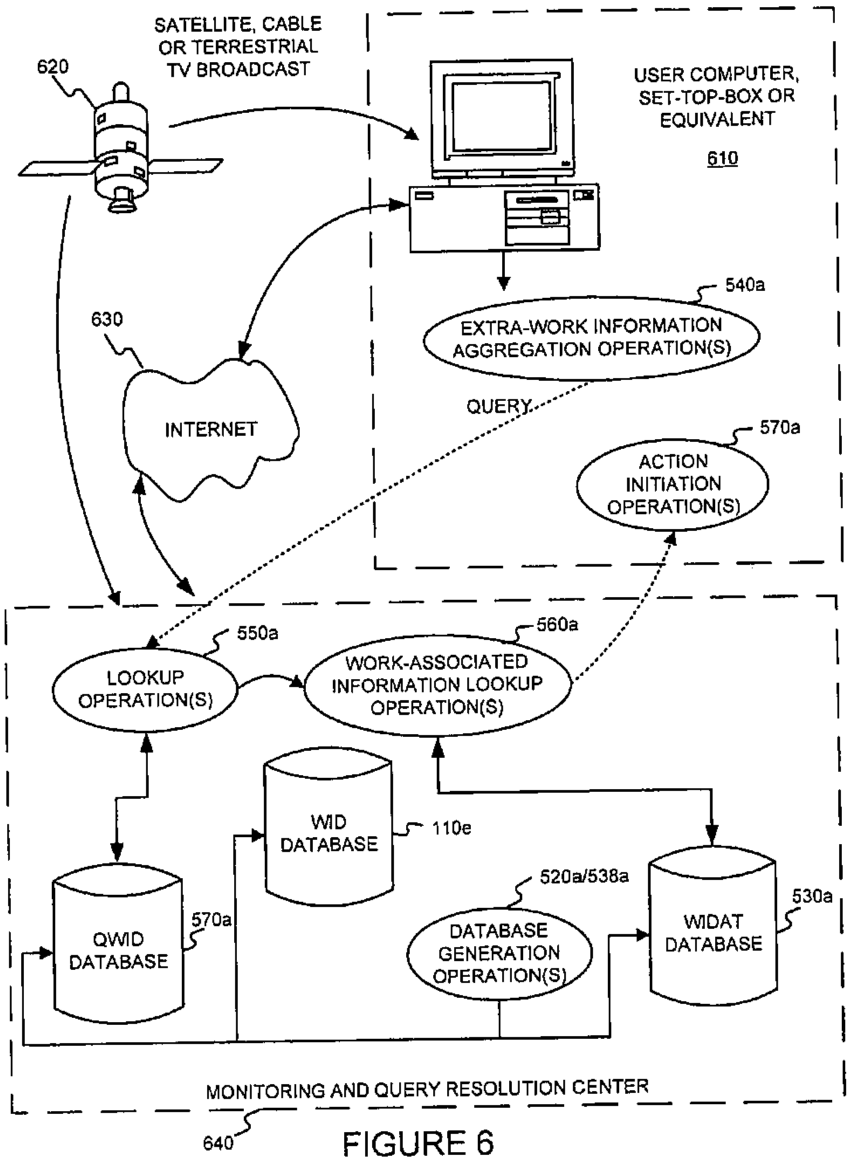

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
620
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
610
540a
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
630
INTERNET
QUERY
570a
ACTION INITIATION OPERATION(S)
550a
LOOKUP OPERATION(S)
560a
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WID DATABASE
110e
520a/538a
DATABASE GENERATION OPERATION(S)
WIDAT DATABASE
530a
QWID DATABASE
570a
MONITORING AND QUERY RESOLUTION CENTER
640

FIGURE 6

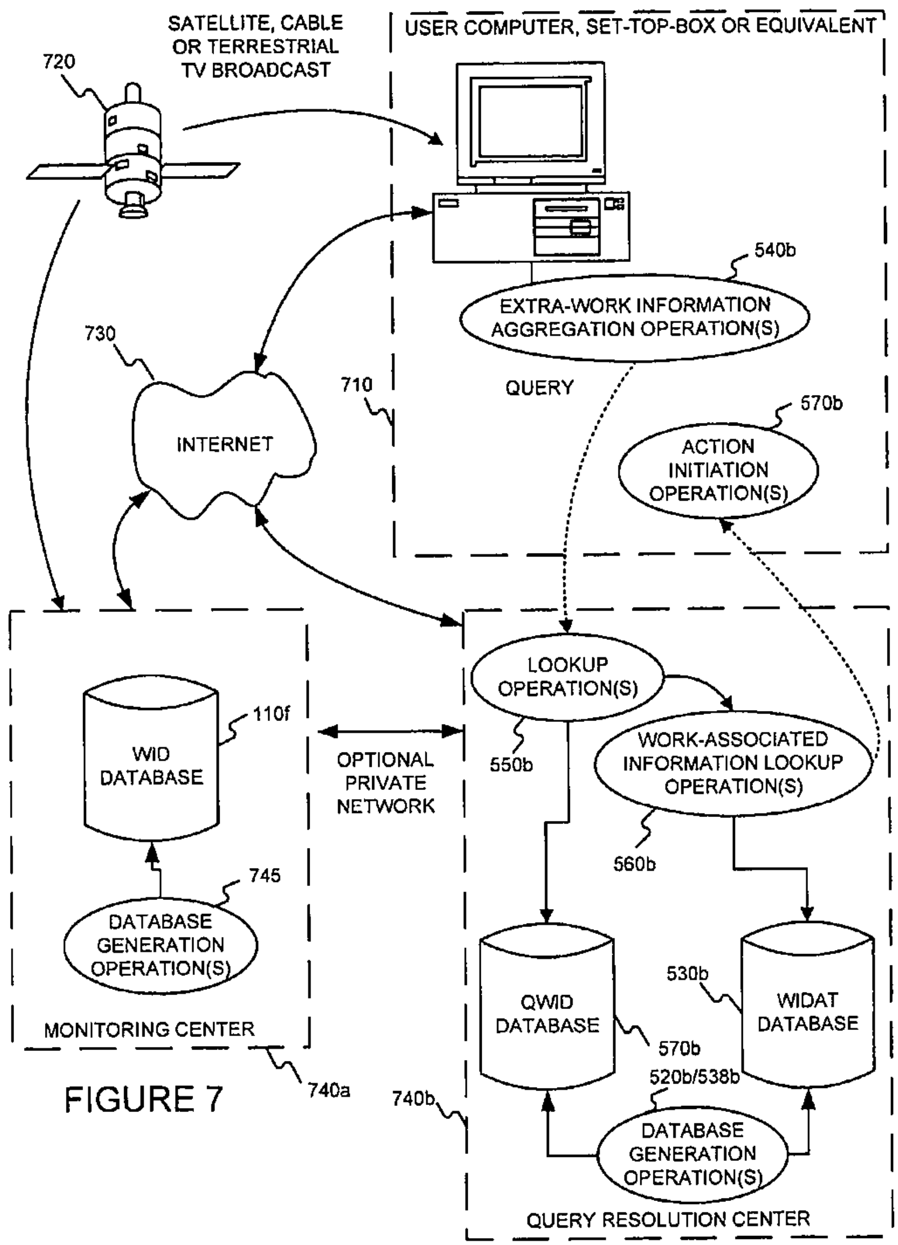

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
720
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
540b
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
730
INTERNET
710
QUERY
570b
ACTION INITIATION OPERATION(S)
LOOKUP OPERATION(S)
550b
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
560b
WID DATABASE
110f
OPTIONAL PRIVATE NETWORK
745
DATABASE GENERATION OPERATION(S)
MONITORING CENTER
740a
QWID DATABASE
570b
530b
WIDAT DATABASE
520b/538b
740b
DATABASE GENERATION OPERATION(S)
QUERY RESOLUTION CENTER

FIGURE 7

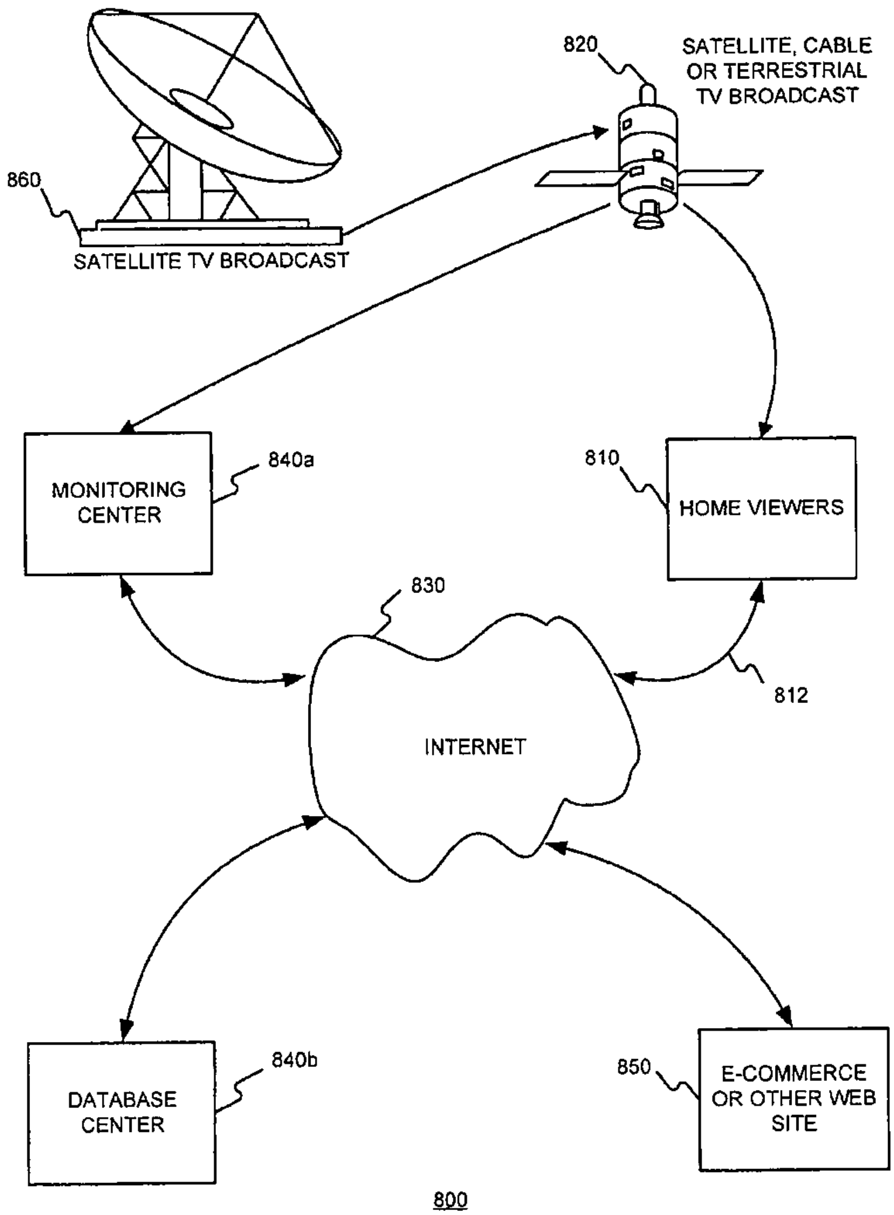

860
SATELLITE TV BROADCAST
820
SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
MONITORING CENTER
840a
810
HOME VIEWERS
830
812
INTERNET
DATABASE CENTER
840b
850
E-COMMERCE OR OTHER WEB SITE
800

FIGURE 8

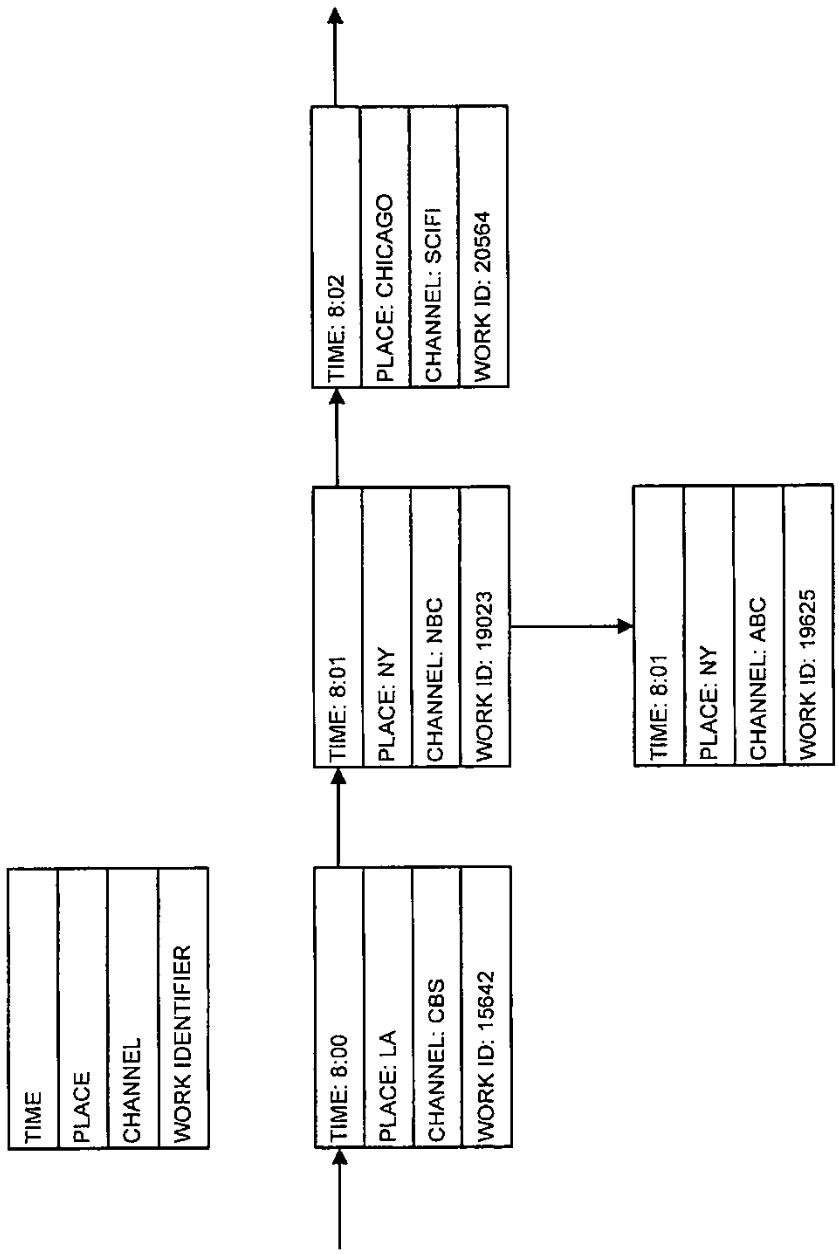

TIME
PLACE
CHANNEL
WORK IDENTIFIER
TIME: 8:00
PLACE: LA
CHANNEL: CBS
WORK ID: 15642
TIME: 8:01
PLACE: NY
CHANNEL: NBC
WORK ID: 19023
TIME: 8:01
PLACE: NY
CHANNEL: ABC
WORK ID: 19625
TIME: 8:02
PLACE: CHICAGO
CHANNEL: SCIFI
WORK ID: 20564

FIGURE 9

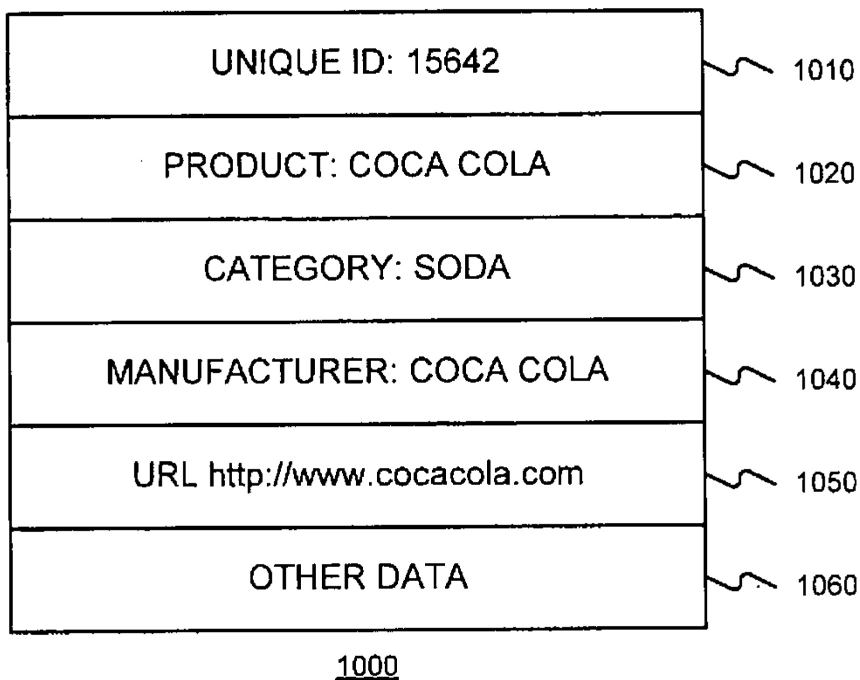

UNIQUE ID: 15642
PRODUCT: COCA COLA
CATEGORY: SODA
MANUFACTURER: COCA COLA
URL http://www.cocacola.com
OTHER DATA
1010
1020
1030
1040
1050
1060
1000

FIGURE 10

USING FEATURES EXTRACTED FROM AN AUDIO AND/OR VIDEO WORK TO OBTAIN INFORMATION ABOUT THE WORK

§0. RELATED APPLICATIONS

The present application is a continuation-in-part of U.S. patent application Ser. No. 09/950,972 (incorporated herein by reference), titled "IDENTIFYING WORKS FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET," filed on Sep. 13, 2001 now U.S. Pat. No. 7,058,223, and listing Ingemar J. Cox as the inventor, which application claims benefit to the filing date of provisional patent application Ser. No. 60/232,618 (incorporated herein by reference), titled "Identifying and linking television, audio, print and other media to the Internet", filed on Sep. 14, 2000 and listing Ingemar J. Cox as the inventor.

§1. BACKGROUND OF THE INVENTION

§1.1 Field of the Invention

The present invention concerns linking traditional media to new interactive media, such as that provided over the Internet for example. In particular, the present invention concerns identifying a work (e.g., content or an advertisement delivered via print media, or via a radio or television broadcast) without the need to modify the work.

§1.2 Related Art

§1.2.1 Opportunities Arising from Linking Works Delivered Via Some Traditional Media Channel or Conduit to a More Interactive System

The rapid adoption of the Internet and associated World Wide Web has recently spurred interest in linking works, delivered via traditional media channels or conduits, to a more interactive system, such as the Internet for example. Basically, such linking can be used to (a) promote commerce, such as e-commerce, and/or (b) enhance interest in the work itself by facilitating audience interaction or participation. Commerce opportunities include, for example, facilitating the placement of direct orders for products, providing product coupons, providing further information related to a product, product placement, etc.

In the context of e-commerce, viewers could request discount vouchers or coupons for viewed products that are redeemable at the point of purchase. E-commerce applications also extend beyond advertisements. It is now common for television shows to include product placements. For example, an actor might drink a Coke rather than a Pepsi brand of soda, actors and actresses might wear designer-labeled clothing such as Calvin Klein, etc. Viewers may wish to purchase similar clothing but may not necessarily be able to identify the designer or the particular style directly from the show. However, with an interactive capability, viewers would be able to discover this and other information by going to an associated Web site. The link to this Web site can be automatically enabled using the invention described herein.

In the context of facilitating audience interaction or participation, there is much interest in the convergence of television and computers. Convergence encompasses a very wide range of capabilities. Although a significant effort is being directed to video-on-demand applications, in which there is a unique video stream for each user of the service, as well as to transmitting video signals over the Internet, there is also interest in enhancing the television viewing experience. To this end, there have been a number of experiments with interactive television in which viewers can participate in a live broadcast. There are a variety of ways in which viewers can participate. For example, during game shows, users can answer the questions and their scores can be tabulated. In recent reality-based programming such as the ABC television game show, "Big Brother", viewers can vote on contestants who must leave the show, and be eliminated from the competition.

§1.2.2 Embedding Work Identifying Code or Signals within Works

Known techniques of linking works delivered via traditional media channels to a more interactive system typically require some type of code, used to identify the work, to be inserted into the work before it is delivered via such traditional media channels. Some examples of such inserted code include (i) signals inserted into the vertical blanking interval ("VBI") lines of a (e.g., NTSC) television signal, (ii) watermarks embedded into images, (iii) bar codes imposed on images, and (iv) tones embedded into music.

The common technical theme of these proposed implementations is the insertion of visible or invisible signals into the media that can be decoded by a computer. These signals can contain a variety of information. In its most direct form, the signal may directly encode the URL of the associated Web site. However, since the alphanumeric string has variable length and is not a particularly efficient coding, it is more common to encode a unique ID. The computer then accesses a database, which is usually proprietary, and matches the ID with the associated web address. This database can be considered a form of domain name server, similar to those already deployed for network addresses. However, in this case, the domain name server is proprietary and the addresses are unique ID's.

There are two principal advantages to encoding a proprietary identifier into content. First, as previously mentioned, it is a more efficient use of the available bandwidth and second, by directing all traffic to a single Web site that contains the database, a company can maintain control over the technology and gather useful statistics that may then be sold to advertisers and publishers.

As an example of inserting signals into the vertical blanking interval lines of a television signal, RespondTV of San Francisco, Calif. embeds identification information into the vertical blanking interval of the television signal. The VBI is part of the analog video broadcast that is not visible to television viewers. For digital television, it may be possible to encode the information in, for example, the motion picture experts group ("MPEG") header. In the USA, the vertical blanking interval is currently used to transmit close-captioning information as well as other information, while in the UK, the VBI is used to transmit teletext information. Although the close captioning information is guaranteed to be transmitted into the home in America, unfortunately, other information is not. This is because ownership of the vertical blanking interval is disputed by content owners, broadcasters and local television operators.

As an example of embedding watermarks into images, Digimarc of Tualatin, OR embeds watermarks in print media. Invisible watermarks are newer than VBI insertion, and have the advantage of being independent of the method of broadcast. Thus, once the information is embedded, it should remain readable whether the video is transmitted in NTSC, PAL or SECAM analog formats or newer digital formats. It should be more reliable than using the vertical blanking interval in television applications. Unfortunately, however, watermarks still require modification of the broadcast signal which is problematic for a number of economic, logistical, legal (permission to alter the content is needed) and quality control (the content may be degraded by the addition of a watermark) reasons.

As an example of imposing bar codes on images, print advertisers are currently testing a technology that allows an advertisement to be shown to a camera, scanner or bar code reader that is connected to a personal computer ("PC"). The captured image is then analyzed to determine an associated Web site that the PC's browser then accesses. For example, GoCode of Draper, UT embeds small two-dimensional bar codes for print advertisements. The latter signal is read by inexpensive barcode readers that can be connected to a PC. AirClic of Blue Bell, Pa. provides a combination of barcode and wireless communication to enable wireless shopping through print media. A so-called "CueCat" reads bar codes printed in conjunction with advertisements and articles in Forbes magazine. Similar capabilities are being tested for television and audio media.

Machine-readable bar codes are one example of a visible signal. The advantage of this technology is that it is very mature. However, the fact that the signal is visible is often considered a disadvantage since it may detract from the aesthetic of the work delivered via a traditional media channel or conduit.

As an example of embedding tones into music, Digital Convergence of Dallas, Tex. proposes to embed identification codes into audible music tones broadcast with television signals.

All the foregoing techniques of inserting code into a work can be categorized as active techniques in that they must alter the existing signal, whether it is music, print, television or other media, such that an identification code is also present. There are several disadvantages that active systems share. First, there are aesthetic or fidelity issues associated with bar codes, audible tones and watermarks. More importantly, all media must be processed, before it is delivered to the end user, to contain these active signals. Even if a system is enthusiastically adopted, the logistics involved with inserting bar codes or watermarks into, say every printed advertisement, are formidable.

Further, even if the rate of adoption is very rapid, it nevertheless remains true that during the early deployment of the system, most works will not be tagged. Thus, consumers that are early-adopters will find that most media is not identified. At best, this is frustrating. At worst, the naïve user may conclude that the system is not reliable or does not work at all. This erroneous conclusion might have a very adverse effect on the adoption rate.

Further, not only must there be modification to the production process, but modifications must also be made to the equipment in a user's home. Again, using the example of watermarking of print media, a PC must be fitted with a camera and watermark detection software must be installed. In the case of television, the detection of the identification signal is likely to occur at the set-top-box—this is the equipment provided by the local cable television or satellite broadcasting company. In many cases, this may require modifications to the hardware, which is likely to be prohibitively expensive. For example, the audible tone used by Digital Convergence to recognize television content, must be fed directly into a sound card in a PC. This requires a physical connection between the television and the PC, which may be expensive or at least inconvenient, and a sound card may have to be purchased.

§1.2.3 Unmet Needs

In view of the foregoing disadvantages of inserting an identification code into a work, thereby altering the existing signal, there is a need for techniques of identifying a work without the need of inserting an identification code into a work. Such an identification code can then be used to invoke a work-related action, such as work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§2. SUMMARY OF THE INVENTION

This patent application describes an alternative solution that does not suffer from the problems outlined above. The solution is based on direct or indirect recognition of the media itself. Direct or indirect recognition refers to the fact that a number of possible configurations are possible, some of which directly recognize the work on the equipment in a user's home while other configurations perform this recognition indirectly by transmitting work-specific information to one or more remote sites. Neither technique requires the embedding of any form of active signal. Instead, when media in the form of music, print, television or multimedia is presented to a personal computer (PC), set-top-box or other device, such devices directly or indirectly recognize the media and initiate an action. The set of possible actions is potentially infinite and includes, for example, retrieving further information, interacting with a live broadcast, registering the user for a service or product, purchasing a product or service and/or receiving discount coupons or certificates that can be used towards a purchase.

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable programs for linking a media work to an action. Such embodiments might (a) extract features from the media work, (b) determine an identification of the media work based on the features extracted, and (c) determine an action based on the identification of the media work determined. In some embodiments consistent with the present invention, the media work is an audio signal. The audio signal might be obtained from a broadcast, or an audio file format. In other embodiments consistent with the present invention, the media work is a video signal. The video signal might be obtained from a broadcast, or a video file format.

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable program for providing information about an audio (or video) file played on a device. Such embodiments might (a) extract features from the audio (or video) file, (b) communicate the features to a database, and (c) receive the information about the audio (or video) file from the database. In some embodiments consistent with the present invention, the act of extracting the features is performed by a microprocessor of the device, and/or a digital signal processor of the device.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§3. BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work.

FIG. 2 is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 3 is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 4 is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 5 is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work.

FIG. 6 is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 8 is a block diagram illustrating an environment in which the present invention may operate.

FIG. 9 is an exemplary data structure in which extra-work information is associated with a work identifier.

FIG. 10 is an exemplary data structure including work-related actions.

§4. DETAILED DESCRIPTION

The present invention may involve novel methods, apparatus and data structures for identifying works without the need of embedding signals therein. Once identified, such information can be used to determine a work-related action. The following description is presented to enable one skilled in the art to make and use the invention, and is provided in the context of particular embodiments and methods. Various modifications to the disclosed embodiments and methods will be apparent to those skilled in the art, and the general principles set forth below may be applied to other embodiments, methods and applications. Thus, the present invention is not intended to be limited to the embodiments and methods shown and the inventors regard their invention as the following disclosed methods, apparatus, data structures and any other patentable subject matter to the extent that they are patentable.

§4.1 FUNCTIONS

The present invention functions to identify a work without the need of inserting an identification code into a work. The present invention may do so by (i) extracting features from the work to define a feature vector, and (ii) comparing the feature vector to feature vectors associated with identified works. Alternatively, or in addition, the present invention may do so by (i) accepting extra-work information, such as the time of a query or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and (ii) use such extra-work information to lookup an identification of the work. In either case, an identification code may be used to identify the work.

The present invention may then function to use such an identification code to initiate a work-related action, such as for work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§4.2 EMBODIMENTS

As just introduced in §4.1 above, the present invention may use intra-work information and/or extra-work information to identify a work. Once identified, such identification can be used to initiate an action, such as an action related to commerce, or facilitating audience participation or interaction. Exemplary embodiments of the present invention, in which work is recognized or identified based on intra-work information, are described in §4.2.1. Then, exemplary embodiments of the present invention, in which work is recognized or identified based on extra-work information, are described in §4.2.2.

§4.2.1 Embodiments in which Work is Recognized Based on Intra-Work Information, Such as a Feature Vector

Operations related to this embodiment are described in §4.2.1.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.1.2.

§4.2.1.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work. As shown, a work-identification information storage **110** may include a number of items or records **112**. Each item or record **112** may associate a feature vector of a work **114** with a, preferably unique, work identifier **116**. The work-identification information storage **110** may be generated by a database generation operation(s) **120** which may, in turn, use a feature extraction operation(s) **122** to extract features from a work at a first time ($WORK_{@t1}$), as well as a feature-to-work identification tagging operation(s) **124**.

Further, work identifier-action information storage **130** may include a number of items or records **132**. Each item or record **132** may associate a, preferably unique, work identifier **134** with associated information **136**, such as an action for example. The work identifier-action information storage **130** may be generated by a database generation operation(s) **138** which may, for example, accept manual entries.

As can be appreciated from the foregoing, the work-information storage **110** records **112** and the work identification-action **130** records **132** can be combined into a single record. That is, there need not be two databases. A single database is also possible in which the work identifier, or a feature vector extracted from the work, serves as a key and the associated field contains work-related information, such as a URL for example.

The feature extraction operation(s) **140** can accept a work, such as that being rendered by a user, at a second time ($WORK_{@t2}$), and extract features from that work. The extracted features may be used to define a so-called feature vector.

The extracted features, e.g., as a feature vector, can be used by a feature (vector) lookup operation(s) **150** to search for a matching feature vector 114. If a match, or a match within a predetermined threshold is determined, then the associated work identifier 116 is read.

The read work identifier can then be used by a work-associated information lookup operation(s) 160 to retrieve associated information, such as an action, 136 associated with the work identifier. Such information 136 can then be passed to action initiation operation(s) 170 which can perform some action based on the associated information 136.

§4.2.1.1.1 Exemplary Techniques for Feature Extraction

When the user initiates a request, the specific television or radio broadcast or printed commercial, each of which is referred to as a work, is first passed to the feature extraction operation. The work may be an image, an audio file or some portion of an audio signal or may be one or more frames or fields of a video signal, or a multimedia signal. The purpose of the feature extraction operation is to derive a compact representation of the work that can subsequently be used for the purpose of recognition. In the case of images and video, this feature vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n×n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, wavelet and or discrete cosine transforms. It might involve principal component analysis. It might also be a combination of these. For television and audio signals, recognition might also rely on a temporal sequence of feature vectors. The recognition literature contains many different representations. For block-based methods, blocks may be accessed at pseudo-random locations in each frame or might have a specific structure. For audio, common feature vectors are based on Fourier frequency decompositions, but other representations are possible. See, e.g., R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley-Interscience, New York, 1973). See also K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Ed. (Academic Press, New York, 1990). (These references are incorporated herein by reference.)

As previously stated, one object of the vector extraction stage is to obtain a more concise representation of the frame. For example, each video frame is initially composed of 480×720 pixels which is equivalent to 345,600 pixels or 691,200 bytes. In comparison, an exemplary feature vector might only consist of 1 Kbyte of data.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e., horizontal and or vertical translation, or may undergo lossy compression such as by MPEG-2. It is advantageous that these and other processes do not adversely affect the extracted vectors. For still images there has been considerable work on determining image properties that are invariant to affine and other geometric distortions. For example, the use of Radon and Fourier-Mellin transforms have been proposed for robustness against rotation, scale and translation, since these transforms are either invariant or bare a simple relation to the geometric distortions. See, e.g., C. Lin, M. Wu, Y. M. Lui, J. A. Bloom, M. L. Miller, I. J. Cox, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," *IEEE Transactions on Image Processing* (2001). See also, U.S. Pat. Nos. 5,436,653, 5,504,518, 5,582,246, 5,612,729, and 5,621,454. (Each of these references is incorporated herein by reference.)

§4.2.1.1.2 Exemplary Techniques for Database Generation and Maintenance

A number of possibilities exist for generating and maintaining work identification (WID) and identification-action translation (WIDAT) databases. However, in all cases, works of interest are processed to extract a representative feature vector and this feature vector is assigned a unique identifier. This unique identifier is then entered into the work identification (WID) database 110 as well as into the WIDAT database 130 together with all the necessary associated data. This process is referred to as tagging. For example, in the case of an advertisement, the WIDAT database 130 might include the manufacturer (Ford), the product name (Taurus), a product category (automotive) and the URL associated with the Ford Taurus car together with the instruction to translate the query into the associated URL.

The determination of all works of interest and subsequent feature vector extraction and tagging depends on whether content owners are actively collaborating with the entity responsible for creating and maintaining the database. If there is no collaboration, then the database entity must collect all works of interest and process and tag them. While this is a significant effort, it is not overwhelming and is certainly commercially feasible. For example, competitive market research firms routinely tabulate all advertisements appearing in a very wide variety of print media. Newspapers and magazines can be scanned in and software algorithms can be applied to the images to identify likely advertisements. These possible advertisements can then be compared with advertisements already in the WID database 110. If there is a match, nothing further need be done. If there is not a match, the image can be sent to a human to determine if the page does indeed contain an advertisement. If so, the operator can instruct the computer to extract the representative feature vector and assign it a unique identifier. Then, the operator can insert this information into the content identification database and as well as update the corresponding WIDAT database 130 with all the necessary associated data. This is continually performed as new magazines and papers include new advertisements to maintain the databases. This is a cost to the database entity. Television and radio broadcasts can also be monitored and, in fact, broadcast monitoring is currently performed by companies such as Nielsen Media research and Competitive Media Reporting. Television and radio broadcasts differ from print media in the real-time nature of the signals and the consequent desire for real-time recognition.

In many cases, advertisers, publishers and broadcasters may wish to collaborate with the database provider. In this case, feature extraction and annotation and/or extra-work information may be performed by the advertiser, advertisement agency, network and/or broadcaster and this information sent to the database provider to update the database. Clearly, this arrangement is preferable from the database provider's perspective. However, it is not essential.

§4.2.1.1.3 Exemplary Techniques for Matching Extracted Features with Database Entries

The extracted feature vector is then passed to a recognition (e.g., feature look-up) operation, during which, the vector is compared to entries of known vectors 114 in a content identification (WID) database 110. It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search might not be possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used including mutual information, Euclidean distance and Lp-norms. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and false negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match. See, e.g., R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley-Interscience, New York, 1973). See also, U.S. Pat. No. 3,919,474 by W. D. Moon, R. J. Weiner, R. A. Hansen and R. N. Linde, entitled "Broadcast Signal Identification System". (Each of these references is incorporated herein by reference.)

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. Unfortunately, binary search is not possible when taking a noisy signal and trying to find the most similar reference signal. This problem is one of nearest neighbor search in a (high-dimensional) feature space. In previous work, it was not uncommon to perform a linear search of all N entries, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this search can be computationally very expensive.

Other forms of matching include those based on clustering, kd-trees, vantage point trees and excluded middle vantage point forests are possible and will be discussed in more detail later. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", *Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop*, (Jan. 15, 1999). See also, P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.)

If the extracted vector "matches" a known vector in the content identification database, then the work has been identified. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when the vector extracted from a work is not matched to the database even though the work is present in the database. There are several reasons why a work's feature vector may fail to match a feature vector database entry. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will often contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible.

Finally, there is the case where the observed work is not present in the database. In this case, the work can be sent to an operator for identification and insertion in the database.

§4.2.1.1.4 Exemplary Work Based Actions

Assuming that the work is correctly identified, then the identifier can be used to retrieve associated information from the second work identification-action translation (WIDAT) database **130** that contains information **136** associated with the particular work **134**. This information may simply be a corresponding URL address, in which case, the action can be considered to be a form of network address translation. However, in general, any information about the work could be stored therein, together with possible actions to be taken such as initiating an e-commerce transaction. After looking up the work identifier **134** in the WIDAT database **130**, an action is performed on behalf of the user, examples of which has been previously described.

In addition to using the system to allow audience members of a work to connect to associated sites on the Internet, a number of other uses are possible. First, the work identification database **130** allows competitive market research data to be collected (e.g., the action may include logging an event). For example, it is possible to determine how many commercials the Coca Cola Company in the Chicago market aired in the month of June. This information is valuable to competitors such as Pepsi. Thus, any company that developed a system as described above could also expect to generate revenue from competitive market research data that it gathers.

Advertisers often wish to ensure that they receive the advertising time that was purchased. To do so, they often hire commercial verification services to verify that the advertisement or commercial did indeed run at the expected time. To do so, currently deployed systems by Nielsen and CMR embedded active signals in the advertisement prior to the broadcast. These signals are then detected by remote monitoring facilities that then report back to a central system which commercials were positively identified. See for example U.S. Pat. No. 5,629,739 by R. A. Dougherty entitled "Apparatus and method for injecting an ancillary signal into a low energy density portion of a color television frequency spectrum", U.S. Pat. No. 4,025,851 by D. E. Haselwood and C. M. Solar entitled "Automatic monitor for programs broadcast", U.S. Pat. No. 5,243,423 by J. P. DeJean, D. Lu and R. Weissman, entitled "Spread spectrum digital data transmission over TV video", and U.S. Pat. No. 5,450,122 by L. D. Keene entitled "In-station television program encoding and monitoring system and method". (Each of these patents is incorporated herein by reference.) Active systems are usually preferred for advertisement verification because the required recognition accuracy is difficult to achieve with passive systems. The passive monitoring system described herein supports commercial verification.

§4.2.1.2 Exemplary Architectures

Three alternative architectural embodiments in which the first technique may be employed are now described with reference to FIGS. **2**, **3**, and **4**.

FIG. **2** is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work and in which a audience member device **210**, such as a PC for example, receives and renders a work that is consumed by an audience member (user). At some point, the user may wish to perform a work-specific action such as traversing to an associated Web site. Upon initiation of this request, the computer **210** performs the operations **140***a*, **150***a*, **160***a* and **170***a*, such as those shown in FIG. **1**. To reiterate, these operations include a feature extraction operation(s) **140***a*, feature vector lookup or matching operation(s) **150***a* in connection with items or records **112***a* in a work-identification (WID) database **110***a*. If a matching feature vector **114***a* is found, the work-associated information lookup operation(s) **160***a* can use the associated work identifier **116***a* to accessing a work identification-action translation (WIDAT) database **130***a* to retrieve associated information **136***a*, possibly including determining what action should be performed.

As described above, the two databases might be integrated into a single database. However, conceptually, they are described here as separate.

An example illustrating operations that can occur in the first embodiment of FIG. 1, is now described. Consider a print application, in which say 10,000 advertisements are to be recognized that appear in national newspapers and magazines. If 1 Kbyte is required to store each feature vector then approximately 10 Mbytes of storage will be required for the work identification database **110***a*. Such a size does not represent a serious problem, in either memory or disk space, to present personal computers.

An important issue then becomes recognition rate. While this may be problematic, all the images are two-dimensional—three-dimensional object recognition is not required. Of course, since a low cost camera captures the printed advertisement, there may be a number of geometric distortions that might be introduced together with noise. Nevertheless, the application is sufficiently constrained that adequate recognition rates should be achievable with current state-of-the-art computer vision algorithms. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, Jan. 15, 1999. See also, P. N. Yianilos "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.) Estimates of the size of the WIDAT database **130***a* depend on what associated information (recall fields **136**) is stored. If, for example, only a URL address is needed, about 20 characters can typically represent most URLs. Thus, the size of the WIDAT database **130***a* would be less than 1 Mbyte.

The configuration just described with reference to FIG. 2 places all of the processing and data on each user's local machine **210**. A number of alternative embodiments, in which some or all of the storage and processing requirements are performed remotely, will be described shortly.

As new works are created and made publicly available, the databases residing on a user's local computer become obsolete. Just as the database provider **240** must continually update the databases in order to remain current, there is also a need to update local databases on devices at audience member premises. This update process can be performed over the Internet **230** in a manner very similar to how software is currently upgraded. It is not necessary to download an entirely new database although this is an option. Rather, only the changes need to be transmitted. During this update process, the user's computer **210** might also transmit information to a central monitoring center **240** informing it of which advertisements the computer user has queried. This type of information is valuable to both advertisers and publishers. Of course, care must be taken to ensure the privacy of individual users of the system. However, it is not necessary to know the identity of individual users for the system to work.

FIG. 3 is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work. Although the WIDAT database can be quite small, as illustrated in the exemplary embodiment described above with respect to FIG. 2, there is still the problem of keeping this database current. While periodic updates of the local databases may be acceptable, they become unnecessary if the WIDAT database **130***b* is at a remote location **340**. In this arrangement, illustrated in FIG. 3, after the local computer **310** identifies the work, it sends a query to the remote WIDAT database **130***b*. The query may contain the work identifier. The remote site **340** may then return the associated information **136**. Although the remote WIDAT database **130***b* needs to be updated by the database provider, this can be done very frequently without the need for communicating the updates to the local computers **310**.

The second embodiment is most similar to active systems in which an embedded signal is extracted and decoded and the identifier is used to interrogate a central database. Consequently it has many of the advantages of such systems, while avoiding the need to insert signals into all works. One such advantage, is that the database provider receives real-time information relating to users' access patterns.

The WIDAT database **130***b* might physically reside at more than one location. In such a case, some requests will go to one site, and other requests will go to another. In this way, overloading of a single site by too many users can be avoided. Other load balancing techniques are also applicable.

FIG. 4 is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work. Recall that the WIDAT database may be small relative to that work identification database (WID). As the size of the work recognition (WID) database increases, the foregoing embodiments may become impractical. Consider, for example, a music application in which it is desired to identify 100,000 song titles. If it is again assumed that a 1 Kbyte vector can uniquely represent each song, then on the order of 100 Mbytes is now needed. This size is comparable to large application programs such as Microsoft's Office 2000 suite. Although this still does not represent an inordinate amount of disk space, if this data needs to reside in memory at all times, then very few present machines will have adequate resources. Clearly, at some point, the proposed architectures scales to a point where requirements become impractical. In this case, a further modification to the architecture is possible.

Since the storage and searching of the work-identifier (WID) database require the most computation and storage, it may be more economical to perform these actions remotely. Thus, for example, if a user is playing an MP3 music file and wants to go to a corresponding website, the MP3 file is passed to an operation that determines one or more feature vectors. In the third embodiment, instead of performing the matching locally **410**, the one or more vectors are transmitted to a central site **440** at which is stored the WID and WIDAT databases **110***c* and **130***c* together with sufficiently powerful computers to resolve this request and those of other computer users. This configuration is illustrated in FIG. 4. Similarly, if a user is playing an MPEG or other video file and wants to initiate a work-related action, the video file is passed to an operation **140***c* that extracts one or more feature vectors. The entire video file need not be processed. Rather, it may be sufficient to process only those frames in the temporal vicinity to the users request, i.e., to process the current frame and or some number of frames before and after the current frame, e.g. perhaps 100 frames in all. The extracted feature vector or feature vectors can then be transmitted to a central site **440** which can resolve the request.

After successfully matching the feature vector, the central site **440** can provide the user with information directly, or can direct the user to another Web site that contains the information the user wants. In cases where the recognition is ambiguous, the central site **440** might return information identifying one of several possible matches and allow the user to select the intended one.

The third embodiment is particularly attractive if the cost of extracting the feature vector is small. In this case, it becomes economical to have feature vector extraction **140***c* in digital set-top-boxes and in video recorders **410**. The latter may be especially useful for the new generation of consumer digital video recorders such as those manufactured by TIVO and Replay TV. These devices already have access to the Internet via a phone line. Thus, when someone watching a recorded movie from television reacts to an advertisement, the video recorder would extract one or more feature vectors and transmit them to a central site 440. This site 440 would determine if a match existed between the query vector and the database of pre-stored vectors 110c. If a match is found, the central server 440 would transmit the associated information, which might include a Web site address or an 800 number for more traditional ordering, back to the audience user device 410. Of course, a consumer device 410 such as a digital video recorder might also store personal information of the owner to facilitate online e-commerce. Such a device 410 could store the owner's name, address, and credit card information and automatically transmit them to an on-line store to complete a purchase. Very little user interaction other than to authorize the purchase might be needed. This type of purchasing may be very convenient to consumers.

Another advantage of the third embodiment is that it obviates the need to update local databases while, at the same time, the centrally maintained databases can be kept current with very frequent updating.

§4.2.2 Embodiments in which Work is Recognized Based on Extra-Work Information

Operations related to this embodiment are described in §4.2.2.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.2.2.

If the cost of extracting a feature vector is too large, then the cost of deploying any of the embodiments described in §4.2.1 above may be prohibitive. This is particularly likely in very cost sensitive consumer products, including set-top-boxes and next generation digital VCR's. Acknowledging this fact, a different technique, one that is particularly well suited for broadcasted media such as television and radio as well as to content published in magazines and newspapers, is now described. This technique relies on the fact that a work need not be identified by a feature vector extracted from the work (which is an example of "intra-work information"), but can also be identified by when and where it is published or broadcast (which are examples of "extra-work information").

An example serves to illustrate this point. Consider the scenario in which a viewer sees a television commercial and responds to it. The embodiments described in §4.2.1 above required the user device (e.g., a computer or set-top-box) 210/310/410 to extract a feature vector. Such an extracted vector was attempted to be matched to another feature vector(s), either locally, or at a remote site. In the embodiments using a remote site, if the central site is monitoring all television broadcasts, then the user's query does not need to include the feature vector. Instead, the query simply needs to identify the time, geographic location and the station that the viewer is watching. A central site can then determine which advertisement was airing at that moment and, once again, return the associated information. The same is true for radio broadcasts. Moreover, magazines and newspapers can also be handled in this manner. Here the query might include the name of the magazine, the month of publication and the page number.

§4.2.2.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. 5 is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work. As shown, a query work-identification (QWID) information storage 510 may include a number of items or records 512. Each item or record 512 may associate extra-work information 514, related to the work, with a, preferably unique, work identifier 516. The query work-identification (QWID) information storage 510 may be generated by a database generation operation(s) 520.

Further, work identifier-action information (WIDAT) storage 530 may include a number of items or records 532. Each item or record 532 may associate a, preferably unique, work identifier 534 with associated information 536, such as an action for example. The work identifier-action (WIDAT) information storage 530 may be generated by a database generation operation(s) 538 which may, for example, accept manual entries.

As can be appreciated from the foregoing, the query work-information (QWID) storage 510 records 512 and the work identification-action (WIDAT) storage 530 records 532 can be combined into a single record.

The extra-work information aggregation (e.g., query generation) operation(s) 540 can accept a information related to a work, such as the time of a user request or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and generate a query from such extra-work information.

The query including the extra-work information can be used by a lookup operation(s) 550 to search for a "matching" set of information 514. If a match, or a match within a predetermined threshold is determined, then the associated work identifier 516 is read.

The read work identifier can then be used by a work-associated information lookup operation(s) 560 to retrieve associated information, such as an action, 536 associated with the work identifier. Such information 536 can then be passed to action initiation operation(s) 570 which can perform some action based on the associated information 536.

If the extra-work information of a work is known (in advance), generating the query work identifier (QWID) information 510 is straight-forward. If this were always the case, an intra-work information-based recognition operation would not be needed. However, very often this is not the case. For example, local television broadcasts typically have discretion to insert local advertising, as well as national advertising. Thus, it often is not possible to know in advance when, on what station, and where a particular advertisement will play.

In such instances, a real-time (e.g., centralized) monitoring facility 580 may be used to (i) extract feature vectors from a work, (ii) determine a work identifier 116 from the extracted features, and (iii) communicate one or more messages 590 in which extra-work information (e.g., time, channel, geographic market) 592 is associated with a work identifier 594, to operation(s) 520 for generating query work identification (QWID) information 510.

§4.2.2.1.1 Exemplary Extra-Work Information

In the context of national broadcasts, geographic information may be needed to distinguish between, for example, the ABC television broadcast in Los Angeles and that in New York. While both locations broadcast ABC's programming, this programming airs at different times on the East and West coasts of America. More importantly, the local network affiliates that air ABC's shows have discretion to sell local advertising as well as a responsibility to broadcast the national commercials that ABC sells. In short, the works broadcast by ABC in Los Angeles can be different from that in other geographic locations. Geographic information is therefore useful to distinguish between the different television markets. In some circumstances, geographic information may not be necessary, especially in parts of the world with highly regulated and centralized broadcasting in which there are not regional differences.

§4.2.2.1.2 Exemplary Techniques for Generating Databases

FIG. 5 illustrates a third database **510** referred to as the query to work identification (QWID) database. This database **510** maps the query (e.g., in the form of time, location and channel information) into a unique ID that identifies the perceived work. The QWID **510** and WIDAT **530** databases might not be separate, but for clarity will be considered so. After retrieving the unique work identifier **512** from the QWID database **510**, the identifier can be used to access the WIDAT database **530**. This is discussed in more detail later.

As introduced above, although it appears that this architecture does not require a recognition facility, such a facility may be needed. The feature extraction operation(s) **140*d***, as well as the work identification operation(s) **150*d*** and other databases **110*d***, may be moved to one or more remote sites **580**.

Although TV Guide and other companies provide detailed information regarding what will be broadcast when, these scheduling guides do not have any information regarding what advertisements will air when. In many cases, this information is unknown until a day or so before the broadcast. Even then, the time slots that a broadcaster sells to an advertiser only provide a time range, e.g. 12 pm to 3 pm. Thus it is unlikely that all commercials and aired programming can be determined from TV schedules and other sources prior to transmission. Further, occasionally programming schedules are altered unexpectedly due to live broadcasts that overrun their time slots. This is common in sports events and awards shows. Another example of interrupts to scheduled programming occurs when a particularly important news event occurs.

During transmission, it may therefore be necessary for a central site **580** to determine what work is being broadcast and to update its and/or other's database **520** accordingly based on the work identified **594** and relevant extra-work information **592**. There are a variety of ways that this can be accomplished.

First, it may be economically feasible to manually monitor all television stations that are of interest, and manually update the database with information regarding the work being monitored. In fact, Nielsen used such procedures in the early 1960's for the company to tabulate competitive market data. More than one person can be employed to watch the same channel in order to reduce the error rate. It should be noted that the recent ruling by the FCC that satellite broadcasters such as DirecTV, DishTV and EchoStar can carry local stations significantly reduces the cost of monitoring many geographic markets. Currently, DirecTV, for example, carries the four main local stations in each of the 35 largest markets. Thus, these 4×35=140 channels can all be monitored from a single site **580**. This site would be provided with satellite receivers to obtain the television channels.

Unfortunately, however, humans are error prone and the monitoring of many different stations from many different geographic locations can be expensive. In order to automate the recognition process, a central site **580** could employ a computer-based system to perform automatic recognition. Because the recognition is centralized, only one or a few sites are needed. This is in comparison with the first architecture we described in which a complete recognition system was required in every user's home or premise. This centralization makes it more economic to employ more expensive computers, perhaps even special purpose hardware, and more sophisticated software algorithms. When video frames or clips cannot be identified or are considered ambiguous, this video can be quickly passed to human viewers to identify. Further, it should be possible for the automated recognition system to use additional information such as television schedules, time of day, etc in order to improve its recognition rate.

§4.2.2.1.2 Exemplary Techniques for Generating Queries Based on Extra-Work Information

At the audience member (user) premises, all that is needed is for the device to send a query to a database-server with information that includes extra-work information, such as geographic location, time and channel. Usually, this extra-work information would be transmitted in real-time, while the work (e.g., an advertisement) is being broadcast. However, this is not necessary. If the television does not have access to the Internet, and most TV's do not yet, then an audience member (user) may simply remember or record which channel he or she was viewing at what time. In fact, the user device could store this information for later retrieval by the user. At a convenient later time, the user might access the Internet using a home PC. At this time, he or she can query the database by entering this extra-work information (e.g., together with geographic information) into an application program or a web browser plug-in.

Another possibility is allowing an audience member (user), at the time he or she is consuming (e.g., viewing, reading, listening to, etc.) the work, to enter query information into a handheld personal digital assistant ("PDA") such as a Palm Pilot, so as not to forget it. This information can then be manually transferred to a device connected to a network, or the information can be transferred automatically using, for example, infrared communications or via a physical link such as a cradle. Recently, PDAs also have some wireless networking capabilities built in, and thus might support direct access to the information desired. Further, software is available that allows a Palm Pilot or other PDA to function as a TV remote control device. As such, the PDA already knows the time of day and channel being viewed. It also probably knows the location of the audience member, since most PDA users include their own name and address in the PDA's phonebook and identify it as their own. Thus, with one or a few clicks, an audience member PDA user could bookmark the television content he or she is viewing. If the PDA is networked, then the PDA can, itself, retrieve the associated information immediately. Otherwise, the PDA can transfer this bookmarked data to a networked device, which can then provide access to the central database.

§4.2.2.2 Exemplary Architectures

FIG. 6 is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work. As shown, an extra-work information aggregation operation **540*a*** may be effected on a device **610**, such as a PC, at the audience member (user) premises. The various databases **510*a***, **530*a***, and **110*e***, as well as the database generation operation(s) **520*a*/538*a***, the lookup operation(s) **550*a*** and the work-associated information lookup operation(s) **560*a*** may be provided at one or more centralized monitoring and query resolution centers **640**.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work. This fifth embodiment is similar to the fourth embodiment illustrated in FIG. 6 but here, the monitoring center 740a and query resolution center 740b are separate.

These embodiments have many advantages for television and radio broadcasters who desire to provide Internet links or other action. First, the audience member (user) equipment, whether it is a computer, set-top-box, television, radio, remote control, personal digital assistant (pda), cell phone or other device, does not need to perform any processing of the received signal. As such, there is almost no cost involved to equipment manufacturers.

These last embodiments have some similarity with services such as those provided by the companies Real Names of Redwood City, Calif., America Online ("AOL") and especially iTag from Xenote. The popular press has reported on the difficulties associated with assigning domain names. The simplest of these problems is that almost all the one-word names in the ".com" category have been used. Consequently, domain names can often be difficult to remember. To alleviate this problem, RealNames and AOL provide alternative, proprietary name spaces (AOL calls these keywords). For a fee, a company may register a name with these companies. Thus, rather than type the URL http://www.bell-labs.com, the simple keyword "bell" might be sufficient to access the same Web site. These capabilities are convenient to users. However, these systems are very different from the fourth and fifth embodiments described. First, and foremost, these systems are not designed to identify content. Rather, they are simply alternative network address translation systems based on easily remembered mnemonics which are sold to interested companies. As such, the user is still expected to type in an address, but this address is easier to remember than the equivalent URL. In contrast, while a user may manually enter the information describing the work, the preferred embodiment is for the computer, set-top-box or other device to automatically generate this information. Further, the mapping of keywords to network addresses is an arbitrary mapping maintained by AOL or Real Names. For example, the keyword "bell" might just as reasonably point to the Web site for Philadelphia's Liberty Bell as to Lucent's Bell Labs. In contrast, the query used in the fourth and fifth embodiments is designed to contain all the necessary data to identify the work, e.g. the time, place and television channel during which the work was broadcast. There is nothing arbitrary about this mapping. It should also be pointed out that the proposed system is dynamic—the same work, e.g. a commercial, potentially has an infinite number of addresses depending on when and where it is broadcast. If an advertisement airs 100,000 unique times, then there are 100,000 different queries that uniquely identify it. Moreover, the exemplary query includes naturally occurring information such as time, place, channel or page number. This is not the case for AOL or RealNames, which typically assigns one or more static keywords to the address of a Web site.

Xenote's iTag system is designed to identify radio broadcasts and uses a query similar to that which may be used in the fourth and fifth embodiments, i.e. time and station information. However, the work identification information is not dynamically constructed but is instead based on detailed program scheduling that radio stations must provide it. As such, it suffers from potential errors in scheduling and requires the detailed cooperation of broadcasters. While the fourth and fifth embodiments might choose to use program scheduling information and other ancillary information to aid in the recognition process, they do not exclusively rely on this. The concept of resolving a site name by recognizing the content is absent from the above systems.

§4.2.3 Exemplary Apparatus for Audience Member (User) Premise Device

While personal computers may be the primary computational device at a user's location, it is not essential to use a PC. This is especially true of the embodiments depicted in FIGS. 6 and 7, which do not require the content, e.g. video signal, to be processed. Instead, only a unique set of identification parameters such as time, location and channel are provided to identify the perceived Work. Many forms of devices can therefore take advantage of this configuration.

As previously noted, personal digital assistants (PDAs) can be used to record the identification information. This information can then be transferred to a device with a network communication such as a PC. However, increasingly, PDAs will already have wireless network communication capabilities built-in, as with the Palm VII PDA. These devices will allow immediate communication with the query resolution center and all information will be downloaded to them or they can participate in facilitating an e-commerce transaction. Similarly, wireless telephones are increasingly offering web-enabled capabilities. Consequently, wireless phones could be programmed to act as a user interface.

New devices can also be envisaged, including a universal remote control for home entertainment systems with a LCD or other graphical display and a network connection. This connection may be wireless or the remote control might have a phone jack that allows it to be plugged directly into an existing phone line. As home networks begin to be deployed, such devices can be expected to communicate via an inexpensive interface to the home network and from there to access the Internet.

In many homes, it is not uncommon for a computer and television to be used simultaneously, perhaps in the same room. A person watching television could install a web browser plug-in or applet that would ask the user to identify his location and the station being watched. Then, periodically, every 20 seconds for example, the plug-in would update a list of web addresses that are relevant to the television programs being watched, including the commercials. The audience member would then simply click on the web address of interest to obtain further information. This has the advantage that the viewer does not have to guess the relevant address associated with a commercial and, in fact, can be directed to a more specialized address, such as www.fordvehicles.com/ibv/tausrus2kflash/flash.html, rather than the generic www.ford.com site. Of course, this applet or plug-in could also provide the database entity with information regarding what is being accessed from where and at what time. This information, as noted earlier, is valuable to advertisers and broadcasters. For PC's that have infra-red communication capabilities, it is straightforward to either control the home entertainment center from the PC or for the PC to decode the signals from a conventional remote control. Thus, as a user changes channels, the PC is able to automatically track the channel changes.

Recording devices such as analog VCR's and newer digital recording devices can also be exploited in the embodiments depicted in FIGS. 6 and 7, especially if device also record the channel and time information for the recorded content. When a user initiates a query, the recorded time and channel, rather than the current time and channel, then form part of the identification information.

Digital set-top-boxes are also expected to exploit the capabilities described herein. In particular, such devices will have two-way communication capabilities and may even include cable modem capabilities. Of course, the two-way communication need not be over a television cable. For example, satellite set-top-boxes provide up-link communications via a telephone connection. Clearly, such devices provide a convenient location to enable the services described herein. Moreover, such services can be provided as part of the OpenCable and DOCSIS (data over cable service interface specification) initiatives.

§4.2.4 Information Retrieval Using Features Extracted from Audio and/or Video Works

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable program for providing information about an audio file or (a video file) played on a device. Such embodiments might (a) extract features from the audio (or video) file, (b) communicate the features to a database, and (c) receive the information about the audio (or video) file from the database. In some embodiments consistent with the present invention, the act of extracting the features is performed by a microprocessor of the device, and/or a digital signal processor of the device. The received information might be rendered on an output (e.g., a monitor, a speaker, etc.) of the device. The received information might be stored (e.g., persistently) locally on the device. The information might be stored on a disk, or non-volatile memory.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§4.3 OPERATIONAL EXAMPLES

An example illustrating operations of an exemplary embodiment of the present invention, that uses intra-work information to identify the work, is provided in

§4.3.1. Then, an example illustrating operations of an exemplary embodiment of the present invention, that uses extra-work information to identify the work, is provided in §4.3.2.

§4.3.1 Operational Example where Intra-Work Information is Used to Identify the Work

A generic system for monitoring television commercials is now described. Obviously, the basic ideas extend beyond this specific application.

The process of recognition usually begins by recognizing the start of a commercial. This can be accomplished by looking for black video frames before and after a commercial. If a number of black frames are detected and subsequently a similar number are detected 30 seconds later, then there is a good chance that a commercial has aired and that others will follow. It is also well known than the average sound volume during commercials is higher than that for television shows and this too can be used as an indicator of a commercial. Other methods can also be used. The need to recognize the beginning of a commercial is not essential. However, without this stage, all television programming must be assumed to be commercials. As such, all video frames must be analyzed. The advantage of determining the presence of a commercial is that less video content must be processed. Since the percentage of advertising time is relatively small, this can lead to considerable savings. For example, commercials can be buffered and then subsequently processed while the television show is being broadcast. This reduces the real-time requirements of a system at the expense of buffering, which requires memory or disk space. Of course, for the applications envisioned herein, a real-time response to a user requires real-time processing.

Once it is determined that an advertisement is being broadcast, it is necessary to analyze the video frames. Typically, a compact representation of each frame is extracted. This vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n×n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, Fourier-Mellin, wavelet and or discrete cosine transforms. It might involve principal component analysis or any combination thereof. The recognition literature contains many different representations. For block-based methods, the n×n blocks may be located at pseudo-random locations in each frame or might have a specific structure, e.g. a complete tiling of the frame. The feature vector might then be composed of the pixels in each block or some property of each block, e.g. the average intensity or a Fourier or other decomposition of the block. The object of the vector extraction stage is to obtain a more concise representation of the frame. Each frame is initially composed of 480×720 pixels which is equivalent to 345,600 bytes, assuming one byte per pixel. In comparison, the feature vector might only consist of 1 Kbyte of data. For example, if each frame is completely tiled with 16×16 blocks, then the number of blocks per frame is 345,600/256=1350. If the average intensity of each block constitutes the feature vector, then the feature vector consists of 1350 bytes, assuming 8-bit precision for the average intensity values. Alternatively, 100 16×16 blocks can be pseudo-randomly located on each frame of the video. For each of these 100 blocks, the first 10 DCT coefficients can be determined. The feature vector then consists of the 100×10=1000 DCT coefficients. Many other variations are also possible. In many media applications, the content possesses strong temporal and spatial correlations. If necessary, these correlations can be eliminated or substantially reduced by pre-processing the content with a whitening filter.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e. horizontal and or vertical translation, or may undergo lossy compression such as MPEG-2. It is advantageous, though not essential, that these and other processes do not adversely affect the extracted vectors.

Each frame's feature vector is then compared with a database of known feature vectors. These known vectors have previously been entered into a content recognition database together with a unique identifier. If a frame's vector matches a known vector, then the commercial is recognized. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when a frame's vector is not matched to the database even though the advertisement is present in the database. There are several reasons why a frame's feature vector may fail to match. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible. Finally, there is the case where the observed commercial is not yet present in the database. In this case, it is necessary to store the commercial and pass it (e.g., to a person) for identification and subsequent entry in the database.

It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search is often not possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used, including clustering techniques. See, e.g., the Duda and Hart reference. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match.

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. However, in current advertisement monitoring applications there is no discussion of efficient search methods. Thus, a linear search of all N entries may be performed, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this can be computationally expensive. Consider a situation in which one out of 100,000 possible commercials is to be identified. Each 30-second commercial consists of 900 video frames. If all 900 frames are stored in the database, then N=90,000,000. Even if only every 10^{th} video frame is stored in the database, its size is still nine million. While databases of this size are now common, they rely of efficient search to access entries, i.e., they do not perform a linear search. A binary search of a 90,000,000-item database requires less than 20 comparisons. In contrast, a linear search will require an average of 45,000,000!

With 9 million entries, if each vector is 1 Kbyte, then the storage requirement is 9 Gigabytes. Disk drives with this capacity are extremely cheap at this time. However, if the database must reside in memory due to real-time requirements, then this still represents a substantial memory requirement by today's standards. One reason that the data may need to be stored in memory is because of the real-time requirements of the database. If 10 channels are being simultaneously monitored within each of 50 geographic areas, then there will be 15,000 queries per second to the content recognition database, assuming each and every frame is analyzed. This query rate is low. However, if a linear search is performed then 675 billion comparisons per second will be required. This is an extremely high computational rate by today's standards. Even if only key frames are analyzed, this is unlikely to reduce the computational rate by more than an order of magnitude.

If an advertisement is not recognized, then typically, the remote monitoring system will compress the video and transmit it back to a central office. Here, the clip is identified and added to the database and the remote recognition sites are subsequently updated. Identification and annotation may be performed manually. However, automatic annotation is also possible using optical character recognition software on each frame of video, speech recognition software, close captioning information and other information sources. As these methods improve in accuracy, it is expected that they will replace manual identification and annotation.

The recognition system described can be considered to be a form of nearest neighbor search in a high dimensional feature space. This problem has been very well studied and is known to be very difficult as the dimensionality of the vectors increases. A number of possible data structures are applicable including kd-trees and vantage point trees. These data structures and associated search algorithms organize a N-point dataset (N=90,000,000 in out previous example) so that sub-linear time searches can be performed on average. However, worst-case search times can be considerably longer. Recently, Yianilos proposed an excluded middle vantage point forest for nearest neighbor search. See, e.g., the Yianilos reference. This data structure guarantees sub-linear worst-case search times, but where the search is now for a nearest neighbor within a fixed radius, T. The fixed radius search means that if the database contains a vector that is within X of the query, then there is a match. Otherwise, no match is found. In contrast, traditional vantage point trees will always return a nearest neighbor, even if the distance between the neighbor and the query is very large. In these cases, if the distance between the query and the nearest neighbor exceeds a threshold, then they are considered not to match. This is precisely what the excluded middle vantage point forest implicitly does.

Using an excluded middle vantage point forest, will allow accurate real-time recognition of 100,000 broadcasted advertisements. This entails constructing an excluded middle vantage point forest based on feature vectors extracted from say 90,000,000 frames of video. Of course, using some form of pre-filtering that eliminates a large number of redundant frames or frames that are not considered to be good unique identifiers can reduce this number. One such pre-filter would be to only examine the I-frames used when applying MPEG compression. However, this is unlikely to reduce the work identification database (WID) size by more than one order of magnitude. Assuming 10 channels are monitored in each of 50 geographic regions, then the query rate is 15,000=10×50×30 queries per second.

§4.3.2 Operational Example where Extra-Work Information is Used to Identify the Work

FIG. **8** depicts a satellite television broadcast system **800**, though cable and traditional broadcast modes are also applicable. Block **810** represents audience members (users) watching a TV channel in their home, which also has a connection **812** to the Internet **820**. Other networks are also possible. The satellite broadcasts are also being monitored by one or more television monitoring centers **840***a*. These centers **840***a* may monitor all or a subset of the television channels being broadcast. They are not restricted to monitoring satellite TV broadcasts but may also monitor cable and traditional terrestrial broadcasts. The primary purpose of these monitoring centers **840***a* is to identify the works being broadcasted. Of particular interest are television advertisements. However, other works, or portions thereof, may also be identified. Each time a new segment of a work is identified, the monitoring system or systems **840***a* update one or more database centers **840***b*, informing them of the time, place, channel and identity of the identified segment. The segment may be a complete thirty second commercial or, more likely, updates will occur more frequently, perhaps at a rate of 1 update per second per channel per geographic location. The database center **840***b* updates its database so that queries can be efficiently responded to in sub-linear time.

The database centers **840***b* can use traditional database technology. In general, the query search initiated by an audience member is not a nearest neighbor search but can be a classical textual search procedure such as a binary search. The nearest neighbor search is appropriate for the monitoring sub-system 840*a*. The database centers 840*b* are continually updated as each new advertisement, television show or portion thereof is recognized. Standard updating algorithms can be used. However, random new entries to the database are unlikely. Rather, each new entry, or set of entries, denotes a new time segment that is later than all previously inserted items. As such, each new entry can be appended to the end of the database while still maintaining an ordered data structure that is amenable to binary and other efficient search techniques. If two entries have the same time in their time field, items can be sorted based on secondary fields such as the channel and geographic location, as depicted in FIG. 9. Since the number of such entries will be relatively small compared with the entire database, it may be sufficient to simply create a linear linked list of such entries, as depicted in FIG. 9. Of course, the size of the database is constantly increasing. As such, it may become necessary to have several levels of storage and caching. Given the envisaged application, most user queries will be for recent entries. Thus, the database may keep the last hours worth of entries in memory. If there is one entry per second for each of 100 channels in 100 geographic locations, this would correspond to 3600×100×100=36,000,000 entries which is easily accommodated in main memory. Entries that are older than one hour may be stored on disk and entries older than one week may be archived (e.g., backed up on tape) for example. The entries to this database can include time, location and channel information together with a unique identifier that is provided by the monitoring system. Of course, additional fields for each entry are also possible.

When a user query is received, the time, channel and geographic information are used to retrieve the corresponding unique identifier that is then used to access a second database that contains information associated with the identified work.

An entry **1000** in this second database is depicted in FIG. **10**, which shows that associated with the unique identifier **1010**, the name of a product **1020**, a product category **1030**, the manufacturer **1040** and the commercial's associated web site **1050**. Many other data fields **1060** are also possible. Such additional fields may include fields that indicate what action should be taken on behalf of the requesting user. Example actions include simply redirecting a request to an associated Web site, or initiating an e-commerce transaction or providing an associated telephone number that may be automatically dialed if the querying device is a cell phone or displaying additional information to the user. This database is likely to be updated much less frequently, perhaps only as often as once or twice a day, as batches of new advertisements are added to the system. Alternatively, it might be updated as each new advertisement is added to the system.

An audience member (user) **810** watching a television commercial for example may react to the advertisement by initiating a query to the database center **840*b***. The device whereby the user initiates the query might be a television or set-top-box remote control, or a computer or a wireless PDA or a (WAP-enabled) cell phone or a specialized device. Typically, the query will occur during the airing of the commercial or a shortly thereafter. However, the time between the broadcasting of the advertisement and the time of the associated query is not critical and can, in some instances be much longer. For example, the audience member might bookmark the query information in a device such as a PDA or a specialized device similar to those developed by Xenote for their Itag radio linking. Later, the audience member may transmit the query to the database center **840*b***. This might happen hours or even days later.

The query contains information that the database center **840*b*** uses to identify the work being viewed. This information might include the time and place where the audience member was, together with the channel being viewed. Other identifying information is also possible. The query may also contain additional information that may be used to facilitate the user's transaction and will include the return address of the user. For example, if the user is intending to order a pizza after seeing a Pizza Hut advertisement, the query may also contain personal information including his or her identity, street address and credit card information.

When the database center **840*b*** receives a query, data in the query is used to identify the work and associated information. A number of possible actions are possible at this point. First, the database center **840*b*** may simply function as a form of proxy server, mapping the audience member's initial query into a web address associated with the advertisement. In this case, the audience member will be sent to the corresponding Web site. The database center **840*b*** may also send additional data included in the initial query to this Web site **850** in order to facilitate an e-commerce transaction between the audience member and the advertiser. In some cases, this transaction will not be direct, but may be indirect via a dealer or third party application service provider. Thus, for example, though an advertisement by Ford Motor Company may air nationally, viewers may be directed to different Web sites for Ford dealerships depending on both the audience member's and the dealerships' geographic locations. In other cases, advertisers may have contracted with the database center **840*b*** to provide e-commerce capabilities. This latter arrangement has the potential to reduce the amount of traffic directed over the public Internet, restricting it, instead to a private network associated with the owner of the database center.

If the audience member (user) is not watching live television but is instead watching a taped and therefore time-shifted copy, then additional processes are needed. For the new generation of digital video recorders, irrespective of the recording media (tape or disk), it is likely to be very easy to include information identifying the location of the recorder, as well as the time and channel recorded. Location information can be provided to the recorder during the setup and installation process, for example. Digital video recorders, such as those currently manufactured by TIVO of Alviso, CA or Replay TV of Santa Clara, Calif. have a network connection via telephone, which can then send the query of an audience member to the database center **840*b*** using the recorded rather than the current information.

In cases where query information has not been recorded, it is still possible to initiate a successful query. However, in this case, it may be necessary to extract the feature vector from the work of interest and send this information to the monitoring center **840*a*** where the feature vector can be identified. This form of query is computationally more expensive but the relative number of such queries compared to those sent to the database centers **840*b*** is expected to be small. It should also be noted that the physical separation of the monitoring and database centers, depicted in FIGS. **6** and **7**, is not crucial to operation of the system and simply serves to more clearly separate the different functionality present in the overall system configuration.

Although the implementation architectures described above focus on the television media, it is apparent that the present invention is applicable to audio, print and other media.

§4.4 CONCLUSIONS

None of the embodiments of the invention require modification to the work or content, i.e. no active signal is embedded. Consequently, there is no change to the production processes. More importantly, from a user perspective, deployment of this system need not suffer from poor initial coverage. Provided the database is sufficiently comprehensive, early adopters will have comprehensive coverage immediately. Thus, there is less risk that the consumer will perceive that the initial performance of the deployed system is poor. Further, the present invention permits statistics to be gathered that measure users' responses to content. This information is expected to be very useful to advertisers and publishers and broadcasters.

What is claimed is:

1. A method for associating an electronic work with an action, the electronic work comprising at least one of audio and video, the method comprising:

a) electronically extracting within a portable client device features from the electronic work;

b) transmitting the extracted features from the portable client device to one or more servers;

c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;

d) electronically determining an action based on the identification of the electronic work; and

e) electronically performing the action on the portable client device.

2. A method of claim 1, wherein the identification is based on a non-exhaustive search identifying a neighbor within a fixed radius.

3. The method of claim 1, wherein the non-exhaustive search is sublinear.

4. The method of claim 1, wherein the non-exhaustive search is based on kd-trees.

5. The method of claim 1, wherein the non-exhaustive search is based on vantage point trees.

6. The method of claim 1, wherein the non-exhaustive search is based on excluded middle vantage point forest.

7. The method of claim 1, wherein the electronic work is an audio work.

8. The method of claim 7, wherein the audio work is obtained from at least one of a broadcast and an audio file format.

9. The method of claim 7, wherein the identification includes at least one of a song title, an album title, and a performer name.

10. The method of claim 1, wherein the electronic work is a video work.

11. The method of claim 10, wherein the video work is obtained from at least one of a broadcast and a video file format.

12. The method of claim 10, wherein the identification includes at least one of a title of the video work, a director of the video work, and names of performers in the video work.

13. The method of claim 1, wherein the step of electronically determining the action includes receiving at the portable client device an action based on the identification of the electronic work from the one or more servers.

14. The method of claim 1, wherein the step of electronically extracting the features is performed by at least one of a microprocessor of the portable client device and a digital signal processor of the portable client device.

15. A method for associating an electronic work with an action, the electronic work comprising at least one of audio and video, the method comprising:

a) electronically extracting features from the electronic work;

b) electronically determining an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;

c) electronically determining an action based on the identification of the electronic work; and

d) electronically performing the action.

16. A method of claim 15, wherein the identification is based on a non-exhaustive search identifying a neighbor within a fixed radius.

17. The method of claim 15, wherein the non-exhaustive search is sublinear.

18. The method of claim 15, wherein the non-exhaustive search is based on kd-trees.

19. The method of claim 15, wherein the non-exhaustive search is based on vantage point trees.

20. The method of claim 15, wherein the non-exhaustive search is based on excluded middle vantage point forest.

21. The method of claim 15, wherein the electronic work is an audio work.

22. The method of claim 21, wherein the audio work is obtained from at least one of a broadcast and an audio file format.

23. The method of claim 21, wherein the identification includes at least one of a song title, an album title, and a performer name.

24. The method of claim 15, wherein the electronic work is a video work.

25. The method of claim 24, wherein the video work is obtained from at least one of a broadcast and a video file format.

26. The method of claim 24, wherein the identification includes at least one of a title of the video work, a director of the video work, and names of performers in the video work.

27. The method of claim 15, wherein the action promotes e-commerce.

28. The method of claim 15, wherein the action promotes interaction.

29. The method of claim 1, wherein the action promotes e-commerce.

30. The method of claim 1, wherein the action promotes interaction.

31. The method of claim 15, wherein the action comprises providing and/or displaying additional information in association with the electronic work.

32. The method of claim 31, wherein the additional information is an advertisement.

33. The method of claim 32, wherein the action comprises providing a link to a site on the World Wide Web associated with the advertisement.

34. The method of claim 32, wherein the action comprises electronically registering a user with at least one of a service and a product related to the advertisement.

35. The method of claim 32, wherein the action comprises electronically providing at least one of a coupon and a certificate related to the advertisement.

36. The method of claim 32, wherein the action comprises automatically dialing a telephone number associated with the advertisement.

37. The method of claim 32, wherein the action comprises collecting competitive market research data related to the advertisement.

38. The method of claim 32, wherein the action comprises purchasing a product or service related to the advertisement.

39. The method of claim 32, wherein the action comprises allowing a user to interact with a live broadcast related to the advertisement.

40. The method of claim 1, wherein the action comprises providing and/or displaying additional information in association with the electronic work.

41. The method of claim 40, wherein the additional information is an advertisement.

42. The method of claim 41, wherein the action comprises providing a link to a site on the World Wide Web associated with the advertisement.

43. The method of claim 41, wherein the action comprises electronically registering a user with at least one of a service and a product related to the advertisement.

44. The method of claim 41, wherein the action comprises electronically providing at least one of a coupon and a certificate related to the advertisement.

45. The method of claim 41, wherein the action comprises automatically dialing a telephone number associated with the advertisement.

46. The method of claim 41, wherein the action comprises collecting competitive market research data related to the advertisement.

47. The method of claim 41, wherein the action comprises purchasing a product or service related to the advertisement.

48. The method of claim 41, wherein the action comprises allowing a user to interact with a live broadcast related to the advertisement.

49. The method of claim 40, wherein the electronic work is an audio work and the additional information comprises at least one of a song title, an album title, and a performer name.

50. The method of claim 40, wherein the electronic work is a video work and the additional information comprises at least one of a title of the video work, a director of the video work, and names of performers in the video work.

51. The method of claim 31, wherein the electronic work is an audio work and the additional information comprises at least one of a song title, an album title, and a performer name.

52. The method of claim 31, wherein the electronic work is a video work and the additional information comprises at least one of a title of the video work, a director of the video work, and names of performers in the video work.

* * * * *

US008205237B2

(12) **United States Patent**
Cox

(10) **Patent No.:** **US 8,205,237 B2**
(45) **Date of Patent:** ***Jun. 19, 2012**

(54) **IDENTIFYING WORKS, USING A SUB-LINEAR TIME SEARCH, SUCH AS AN APPROXIMATE NEAREST NEIGHBOR SEARCH, FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET**

(76) Inventor: **Ingemar J. Cox**, London (GB)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 594 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/977,202**

(22) Filed: **Oct. 23, 2007**

(65) **Prior Publication Data**

US 2008/0060036 A1 Mar. 6, 2008

Related U.S. Application Data

(63) Continuation of application No. 11/445,928, filed on Jun. 2, 2006, which is a continuation-in-part of application No. 09/950,972, filed on Sep. 13, 2001, now Pat. No. 7,058,223.

(60) Provisional application No. 60/232,618, filed on Sep. 14, 2000.

(51) **Int. Cl.**
H04N 7/173 (2011.01)

(52) **U.S. Cl.** **725/110**

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,919,479 A	11/1975	Moon et al.
4,230,990 A	10/1980	Lert, Jr. et al.
4,450,531 A	5/1984	Kenyon et al.
4,495,526 A	1/1985	Baranoff-Rossine
4,499,601 A	2/1985	Matthews
4,511,917 A	4/1985	Kohler et al.
4,547,804 A	10/1985	Greenberg
4,634,966 A	1/1987	Nakatani et al.
4,639,779 A	1/1987	Greenberg
4,677,455 A	6/1987	Okajima
4,677,466 A	6/1987	Lert, Jr. et al.
4,682,370 A	7/1987	Matthews
4,697,209 A	9/1987	Kiewit
4,739,398 A	4/1988	Thomas et al.
4,776,017 A	10/1988	Fujimoto
4,805,020 A	2/1989	Greenberg

(Continued)

OTHER PUBLICATIONS

Peter N. Yianilos, Excluded Middle Vantage Point Forest for Nearest Neighbor Search, Aug. 1, 1999, pp. 1-12.*

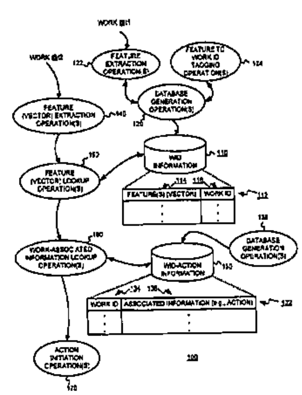
(Continued)

Primary Examiner — Brian Pendleton
Assistant Examiner — Cai Chen
(74) *Attorney, Agent, or Firm* — Amster, Rothstein & Ebenstein LLP

(57) **ABSTRACT**

A media work may be associated with an action by (a) extracting features from the media work, (b) determining an identification of the media work, based on the features extracted, using a sub-linear time search, such as an approximate nearest neighbor search for example, and (c) determining an action based on the identification of the media work determined. The media work may be an audio work. The features extracted from the work may include (A) a frequency decomposition of a signal of the audio work, (B) information samples of the audio work, (C) average intensities of sampled windows of the audio work, and/or (D) information from frequencies of the audio work.

40 Claims, 10 Drawing Sheets



U.S. PATENT DOCUMENTS

4,843,526	A	6/1989	Price, III	
4,843,562	A	6/1989	Kenyon et al.	
4,918,730	A	4/1990	Schulze	
5,210,820	A	5/1993	Kenyon	
5,283,819	A	2/1994	Glick et al.	
5,437,050	A	7/1995	Lamb et al.	
5,481,294	A	1/1996	Thomas et al.	
5,581,658	A	12/1996	O'Hagan et al.	
5,594,934	A	1/1997	Lu et al.	
5,629,739	A	5/1997	Dougherty	
5,692,213	A	11/1997	Goldberg et al.	
5,701,452	A	12/1997	Siefert	
5,701,542	A	12/1997	Sasayama	
5,724,605	A	3/1998	Wissner	
5,745,900	A	4/1998	Burrows	
5,798,785	A	8/1998	Hendricks et al.	
5,850,490	A	12/1998	Johnson	
5,918,223	A	6/1999	Blum et al.	
5,953,415	A	9/1999	Nielsen	
6,006,256	A	12/1999	Zdepski et al.	
6,011,758	A	1/2000	Dockes et al.	
6,026,439	A	2/2000	Chowdhury et al.	
6,044,402	A	3/2000	Jacobson et al.	
6,052,693	A	4/2000	Smith et al.	
6,061,056	A *	5/2000	Menard et al.	715/704
6,088,455	A	7/2000	Logan et al.	
6,088,707	A	7/2000	Bates et al.	
6,118,450	A	9/2000	Proehl et al.	
6,119,124	A	9/2000	Broder et al.	
6,169,986	B1	1/2001	Bowman et al.	
6,173,406	B1	1/2001	Wang et al.	
6,240,409	B1	5/2001	Aiken	
6,243,725	B1	6/2001	Hempleman et al.	
6,247,133	B1	6/2001	Palage et al.	
6,253,193	B1	6/2001	Ginter et al.	
6,263,348	B1	7/2001	Kathrow et al.	
6,330,593	B1	12/2001	Roberts et al.	
6,345,256	B1	2/2002	Milsted et al.	
6,349,296	B1	2/2002	Broder	
6,360,215	B1	3/2002	Judd et al.	
6,363,377	B1	3/2002	Kravets et al.	
6,374,225	B1	4/2002	Hejna, Jr.	
6,381,601	B1	4/2002	Fujiwara et al.	
6,385,596	B1	5/2002	Wiser et al.	
6,408,128	B1	6/2002	Abecassis	
6,418,421	B1	7/2002	Hurtado et al.	
6,446,068	B1	9/2002	Kortge	
6,449,226	B1	9/2002	Kumagai	
6,452,874	B1	9/2002	Otsuka et al.	
6,477,704	B1	11/2002	Cremia	
6,496,802	B1	12/2002	Van Zoest et al.	
6,505,160	B1	1/2003	Levy	
6,550,001	B1	4/2003	Corwin et al.	
6,550,011	B1	4/2003	Sims, III	
6,577,746	B1	6/2003	Evans et al.	
6,591,245	B1	7/2003	Klug	
6,598,228	B2	7/2003	Hejna, Jr.	
6,609,105	B2	8/2003	Van Zoest et al.	
6,654,757	B1	11/2003	Stern	
6,665,661	B1	12/2003	Crow et al.	
6,675,174	B1	1/2004	Bolle et al.	
6,834,308	B1 *	12/2004	Ikezoye et al.	709/231
6,873,982	B1	3/2005	Bates et al.	
6,931,451	B1	8/2005	Logan et al.	
6,941,275	B1	9/2005	Swierczek	
6,978,419	B1	12/2005	Kantrowitz	
6,978,461	B2	12/2005	Shapiro et al.	
6,990,453	B2	1/2006	Wang et al.	
7,013,301	B2	3/2006	Holm et al.	
7,058,223	B2	6/2006	Cox	
7,106,904	B2	9/2006	Shima	
7,155,449	B2	12/2006	Pingel et al.	
7,158,929	B2	1/2007	Wouters et al.	
7,168,083	B2	1/2007	Kalker et al.	
7,302,574	B2	11/2007	Conwell et al.	
7,366,718	B1	4/2008	Pugh et al.	
7,421,723	B2	9/2008	Harkness et al.	
7,477,739	B2	1/2009	Haitsma et al.	
7,523,312	B2	4/2009	Kalker et al.	
7,587,728	B2	9/2009	Wheeler et al.	
7,647,604	B2	1/2010	Ramaswamy	
7,650,616	B2	1/2010	Lee	
7,757,248	B2	7/2010	Harkness et al.	
2001/0001160	A1 *	5/2001	Shoff et al.	725/51
2001/0003818	A1	6/2001	Pingel et al.	
2002/0023020	A1	2/2002	Kenyon et al.	
2002/0032698	A1	3/2002	Cox	
2002/0120925	A1	8/2002	Logan	
2002/0156760	A1	10/2002	Lawrence et al.	
2003/0106017	A1	6/2003	Kanchirayappa et al.	
2003/0146940	A1	8/2003	Ellis	
2004/0199387	A1 *	10/2004	Wang et al.	704/243
2005/0160363	A1	7/2005	Bhogal et al.	
2006/0101069	A1	5/2006	Bell et al.	
2006/0206462	A1	9/2006	Barber	
2007/0041667	A1	2/2007	Cox	
2007/0083510	A1	4/2007	McArdle	
2007/0118375	A1	5/2007	Kenyon et al.	
2008/0091684	A1	4/2008	Ellis et al.	
2008/0250241	A1	10/2008	Ginter et al.	

OTHER PUBLICATIONS

Peter N. Yianlos, Excluded Middle Vantage Point Forest for Nearest Neighbor Search, Aug. 1, 1999, pp. 1-12.*

P.N. Yianilos, "Locally Lifting the Curse of Dimensionality for Nearest Neighbor Search" *SODA 2000*, pp. 361-370.

Baum, L., et al., "A Maximation Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *The Annals of Mathematical Statistics*, vol. 41, No. 1, pp. 164-171 (1970).

Dempster, A. P., et al., "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, Issue 1, pp. 1-38 (1977).

Reynolds, D., et al., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, vol. 3, No. 1, pp. 72-83 (Jan. 1995).

Nievergelt, J. et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," ACM Transactions on Database Systems, vol. 9, No. 1, pp. 38-71 (Mar. 1984).

Heintze, N, "Scalable Document Fingerprinting," Proc. USENIX Workshop on Electronic Commerce (1996).

Wold, E, et al., "Content-Based Classification, Search, and Retrieval of Audio," IEEE Multimedia, vol. 3, Issue 3, pp. 27-63 (1996).

Bhanu, B., et al., "Learning Feature Relevance and Similarity Metrics in Image Databases", Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 14-19 (1998).

Del Bimbo, A., et al., "Using Weighted Spatial Relationships in Retrieval by Visual Contents", Image Description and Retrieval, pp. 161-192 (1998).

Indyk, P., and Motwani, R., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," Proceeding of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604-613 (1998).

La Cascia, M., et al., "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web", Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 24-29 (1998).

Yoshitaka, A., et al., "A Survey on Content-Based Retrieval for Multimedia Databases", IEEE Transactions on Knowledge and Data Engineering, vol. 11, No. 1, pp. 81-93 (Jan./Feb. 1999).

Lawrence, S., et al., "Digital Libraries and Automonous Citation Indexing," IEEE Computer, pp. 67-71 (Jun. 1999).

Kimura, A, et al., "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," IEEE Conf on Acoustics, Speech and Signal Processing, (ICASSP '01), vol. 3, pp. 1429-1432 (2001).

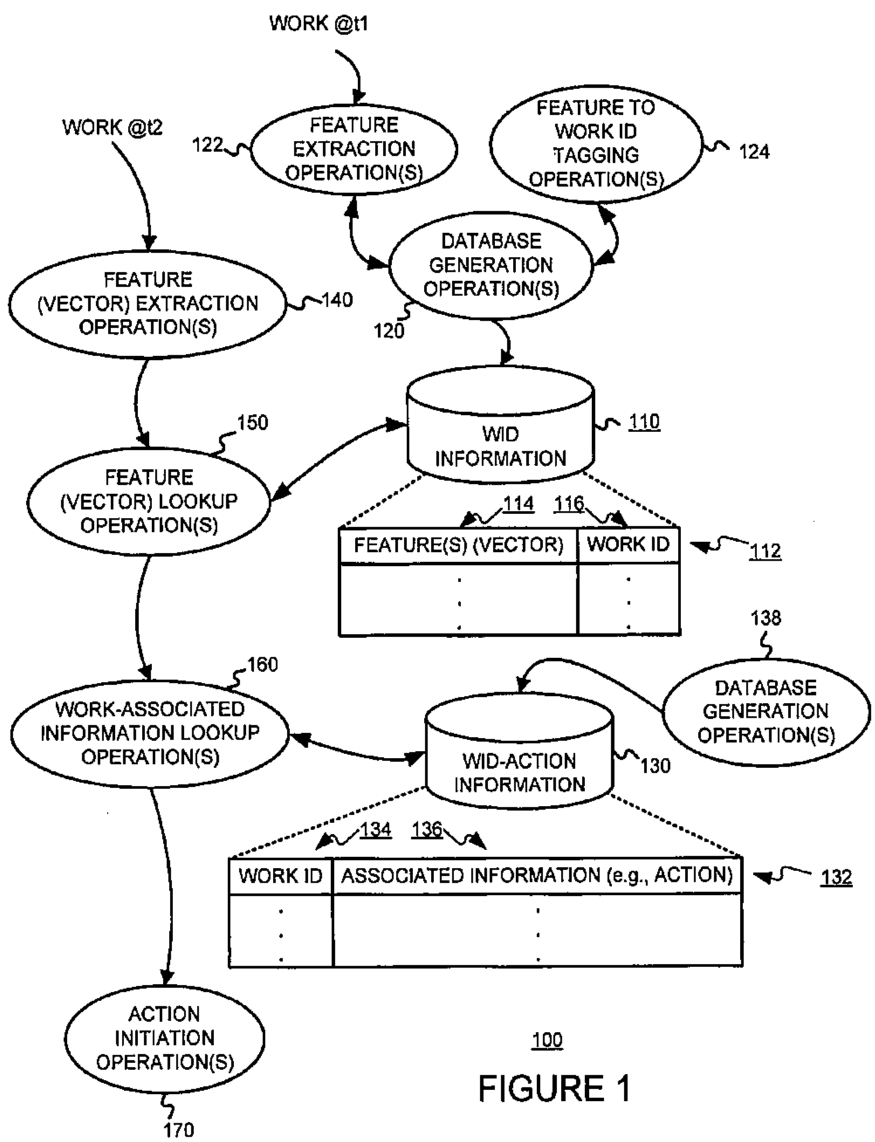
Chavez, E., et al., "Searching in Metric Spaces", ACM Computing Surveys, vol. 33, No. 3, pp. 273-321 (Sep. 2001).

Haitsma, J., et al., "Robust Audio Hashing for Content Identification, Int" Workshop on Content Based Multimedia Indexing, Brescia, Italy (Sep. 19-21, 2001).

Haitsma, J., and Walker, T, "A Highly Robust Audio Fingerprinting System," Journal of New Music Research, 1744-5027, vol. 32, Issue 2, pp. 211-221 (2003).
Schleimer, Saul, et al., "Winnowing: Local Algorithms for Document Fingerprinting ACM SIGMOD" (Jun. 9-12, 2003).
"Searching Near-Replicas of Images via Clustering" Edward Chang, Chen Li, James Wang, Peter Mork, Gio Wiederhold Proc. SPIE Symposium of Voice, Video, and Data Communications, 1999.
"RIME: A Replicated Image Detector for the World-Wide Web" Edward Y. Chang, James Ze Wang, Chen Li, and Gio Wiederhold, SPIE 1998.
"Safeguarding and charging for information on the internet," H. Garcia-Molina, S. Ketchpel, and N. Shivakumar, Proceedings of ICDE, 1998.
"Detection mechanisms for digital documents," S. Brin and H. Garcia-Molina, Proceedings of ACM SIG-MOD, May 1995.
"The x-tree: An index structure for high-dimensional data," S. Berchtold, Proceedings of the 22nd VLDB, Aug. 1996.
"The sr-tree: An index structure for high-dimensional nearest neighbor queries," N. Katayama and S. Satoh, Proceedings of ACM SIGMOD, May 1997.
"The k-d-b-tree: A search structure for large multidimensional dynamic indexes," J. T. Robinson, Proceedings of ACM SIGMOD, Apr. 1981.
"Query by image and video content: The QBIC system," M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and et al, IEEE Computer 28(9), pp. 23{32, 1995.
"Visual information retrieval," A. Gupta and R. Jain, Communications of the ACM 40(5), pp. 69-79, 1997.
"Visualseek: A fully automated content-based image query system," J. R. Smith and S.-F. Chang, ACM Multimedia Conference, 1996.
"Similarity indexing: Algorithms and performance," D. A. White and R. Jain, Proc. SPIE vol. 2670, San Diego, 1996.
"The r*-tree: an efficient and robust access method, for points and rectangles," N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, Proceedings of ACM Sigmod, May 1990.
"R-trees: a dynamic index structure for spatial searching," A. Guttman, Proceedings of ACM Sigmod, Jun. 1984.
"Similarity indexing with the ss-tree," D. A. White and R. Jain, Proceedings of the 12th ICDE, Feb. 1996.
"The tv-tree: an index structure for high-dimensional data," K.-L. Lin, H. V. Jagadish, and C. Faloutsos, VLDB Journal 3 (4), 1994.
"M-tree: An efficient access method for similarity search in metric spaces," P. Ciaccia, M. Patella, and P. Zezula, Proceedings of the 23rd VLDB, Aug. 1997.
"Nearest neighbor queries," N. Roussopoulos, S. Kelley, and F. Vincent, Proceedings of ACM Sigmod, May 1995.
"An extensible hashing index for high-dimensional similarity search," C. Li, E. Chang, and J. Z. Wang, Stanford Technical Report, Aug. 1998.
"Two algorithms for nearest-neighbor search in high dimensions" J. M. Kleinberg, Proc 29th STOC, 1997.
"A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996.
"Adaptive Color Image Embeddings for Database Navigation" Yossi Rubner, Carlo Tomasi and Leonidas J. Guibas, Proceedings of the 1998 IEEE Asian Conference on Computer Vision.
A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces R. Weber, H-J Schek, S. Blott Proc., 24th VLDB Conf. 1998.
Bouktache, D, "A fast algorithm for the nearest neighbor classifier", IEEE Transactions on Pattern Analysis and Machine Intelligence, Mar. 1997, pp. 277-282.
Nene et al., "A simple algorithm for nearest neighbor search in high dimensions", IEEE Transactions on Pattern Analysis and Machine Intelligence; Sep. 1997, pp. 989-1003.
Arya et al. "Approximate nearest neighbor queries in fixed dimensions", Proceedings of the 4th annual ACM-SIAM Symposium on Discrete algorithms, 1993; pp. 271-280.
K. Fukunaga and P. M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. IEEE Trans. Comput., C-24:750{753, Jul. 1975.
C.D. Feustel and L. G. Shapiro. The nearest neighbor problem in an abstract metric space. Pattern Recognition Letters, pp. 125{128, Dec. 1982.
Dennis Shasha and Tsong-Li Wang. New techniques for best-match retrieval. ACM Transactions on Information Systems, 8(2):140{158, Apr. 1990.
J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. Information Processing Letters, 40 (4):175{9, Nov. 1991.
Sergey Brin, "Near Neighbor Search in Large Metric Spaces", Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995.
D. P. Huttenlocher, et al. Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(3):850{63, Sep. 1993.
Seidl et al. "Optimal multi-step k-nearest neighbor search", Proceedings of ACM SIGMOD international conference on Managemet of data, 1998, pp. 154-165.
W.A. Burkhard and R.M. Keller. Some Approaches to Best-Match File Searching. Communications of the ACM. vol. 16, No. 4, Apr. 1973.
Kushilevitz et al. "Efficient search for approximate nearest neighbor in high dimensional spaces", Proceedings of the 30th annual ACM Symposium on Theory of computing, 1998, pp. 614-623.* ;annual ACM Symposium on Theory of computing, 1998, pp. 614-623.
Yianilos, P, "Data structures and algorithms for nearest neighbor search in general metric spaces", Proceedings of the ACM-SIAM Symposium on Discrete algorithms, 1993, pp. 311-321.
Ardizzone, Edoardo et al., "Motion and Color-Based Video Indexing and Retrieval," Universita di palermo, Departimento di Ingegneria Elettrica, pp. 135-139, Viale delle Scienze, Palermo, Italy, IEEE 1996.
Deng, Yining et al., "Content-based Search of Video Using Color, Texture, and Motion," Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA, pp. 534-537, IEEE 1997.
Fang, Min et al., "Computing Iceberg Queries Efficiently," Dept. of Computer Science, Stanford, CA, Paper No. 234, pp. 1-25.
Flickner, Myron et al., "Query by Image and Video Content: The QBIC System," IBM Almaden Research Center, Sep. 1995, pp. 23-32, IEEE 1995.
Gargi, U et al., "Performance Characterization and Comparison of Video Indexing Algorithms," Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA.
Gionis, Aristides et al., "Similarity Search in High Demensions via Hashing," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 518-529, Proceeding the 25th VLDB Conference, Edinburgh, Scotland, 1999.
Indyk, Piotr et al., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality" (preliminary version) Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13 & i-vii, Jul. 21, 1999.
Iyengar, Giridharan et al., "Models for automatic classification of video sequences," MIT Media Laboratory, Cambridge, MA.
Jain, Anil K., et al., "Image Retrieval using Color and Shape," Dept. of Computer Science, Michigan State University, Eas Lansing, MI, pp. 1-24, May 15, 1995.
Ogle, Virginia E., et al., "Chabot: Retrieval from a Relational Database of Images," University of California at Berkeley, Computer pp. 40-48, IEEE 1995.
Pentland, A. et al., "Photobook: Content-Based Manipulation of Image Databases," Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Tech., International Jorunal of Computer Vision 18(3), pp. 233-254 (1996), 1996 Kluwer Academic Publishers. Manuf. in The Netherlands.
Shivakumar, Narayanan et al., "SCAM: A Copy Detection Mechanism for Digital Documents," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13.

Shivakumar, Narayanan et al., "Building a Scalable and Accurate Copy Detection Mechanism," Dept. of Computer Science, Stanford University, Stanford, CA.
Srihari, Rohiini K., "Automatic Indexing and Content-Based Retrieval of Captioned Images," State University of New York, Buffalo, Theme Feature, pp. 49-56, Sep. 1995, IEEE 1995.
Swain, Michael and Ballard, Dana H., "Color Indexing," International Journal of Computer Vision 7:1, p. 11-32 (1991), 1991 Kluwer Academic Publishers. Manuf. in The Netherlands.
Wactlar, Howard D. et al., "Intelligence Access to Digital Video: Informedia Project," Carnegie Mellon University, Digital Library Initiative: Carnegie Mellon University, Computer, pp. 46-52, IEEE 1996.
Yeo, Boon-Lock et al., "Rapid Scene Analysis on Compressed Video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, No. 6, pp. 533-544, Dec. 1995, Dept. of Electrical Engineering, Princeton University, Princeton, NJ, IEEE Log No. 9415901.
Indyk, Piotr et al., "Finding pirated video sequences on the Internet," Dept. of Computer Science, Stanford University, Palo Alto, CA, Paper No. 199.

* cited by examiner



WORK @t1
WORK @t2
FEATURE EXTRACTION OPERATION(S)
122
FEATURE TO WORK ID TAGGING OPERATION(S)
124
DATABASE GENERATION OPERATION(S)
120
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140
WID INFORMATION
110
150
FEATURE (VECTOR) LOOKUP OPERATION(S)
114
116
FEATURE(S) (VECTOR)
WORK ID
112
138
160
DATABASE GENERATION OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WID-ACTION INFORMATION
130
134
136
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
132
ACTION INITIATION OPERATION(S)
170
100

FIGURE 1

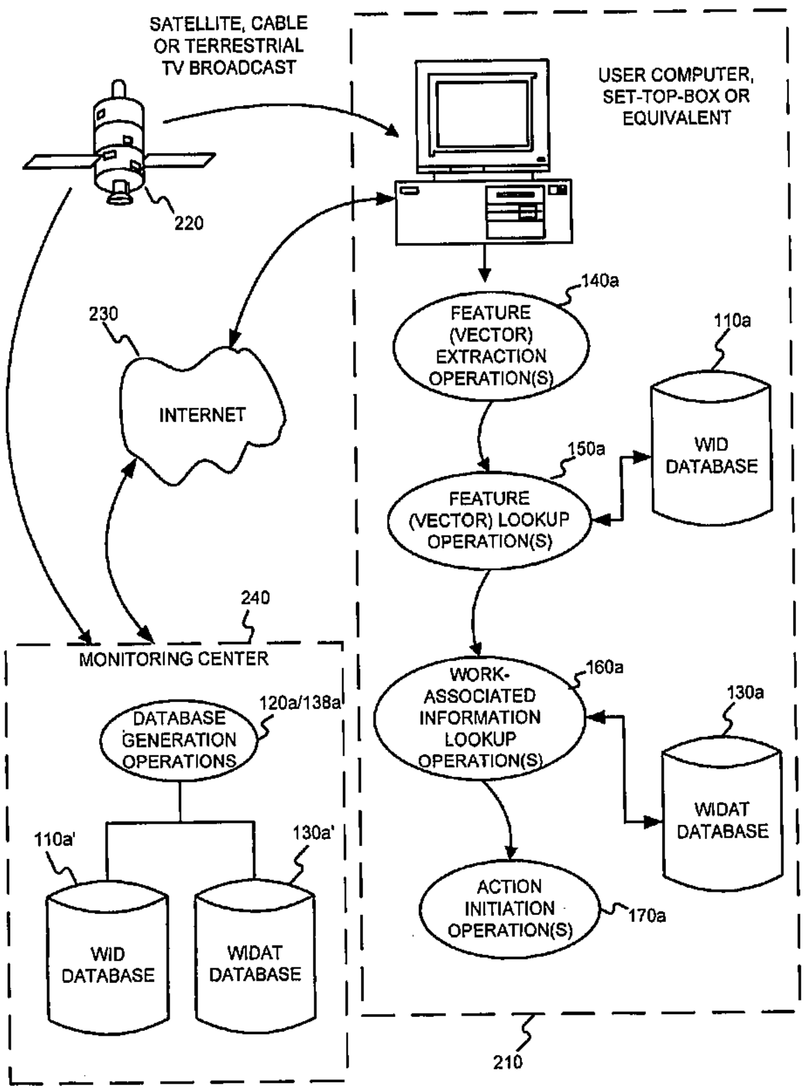

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
220
140a
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
230
110a
INTERNET
WID
DATABASE
150a
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
240
MONITORING CENTER
160a
WORK-
ASSOCIATED
INFORMATION
LOOKUP
OPERATION(S)
130a
DATABASE
GENERATION
OPERATIONS
120a/138a
110a'
130a'
WIDAT
DATABASE
ACTION
INITIATION
OPERATION(S)
170a
WID
DATABASE
WIDAT
DATABASE
210

FIGURE 2

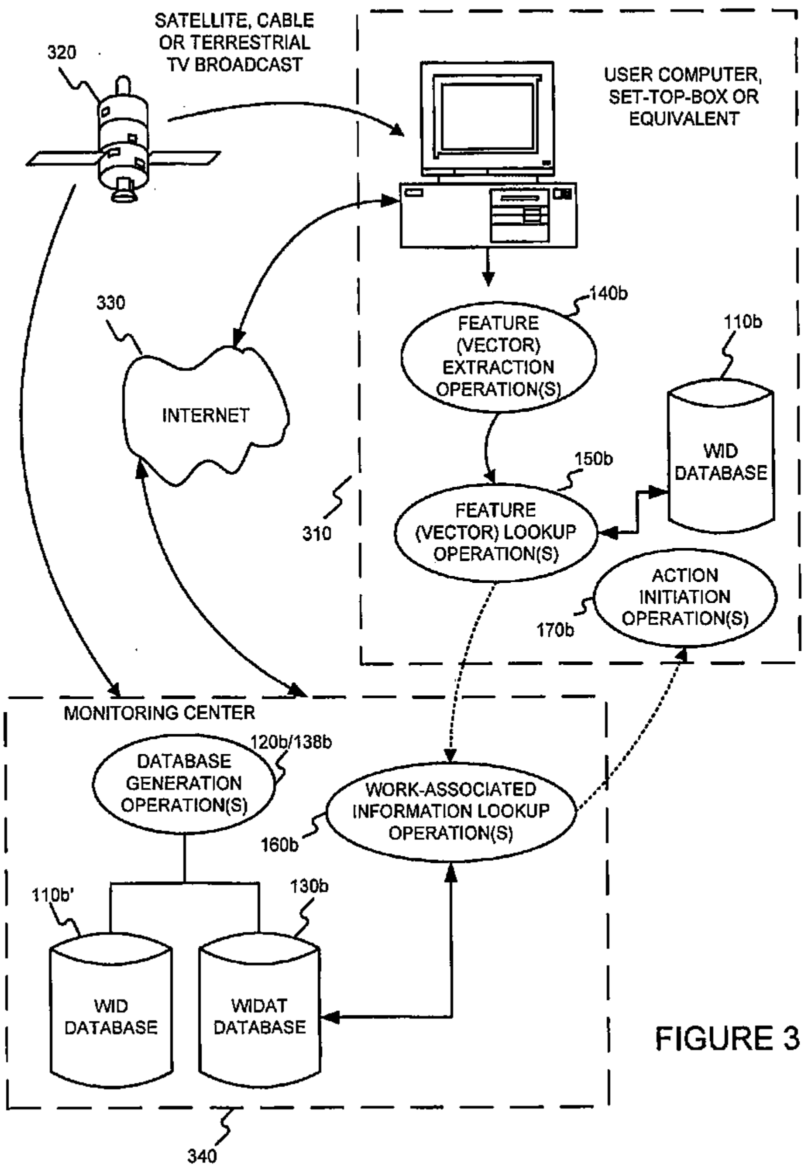

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
320
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
330
INTERNET
140b
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
110b
WID
DATABASE
150b
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
310
ACTION
INITIATION
OPERATION(S)
170b
MONITORING CENTER
120b/138b
DATABASE
GENERATION
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
160b
110b'
130b
WID
DATABASE
WIDAT
DATABASE
340

FIGURE 3

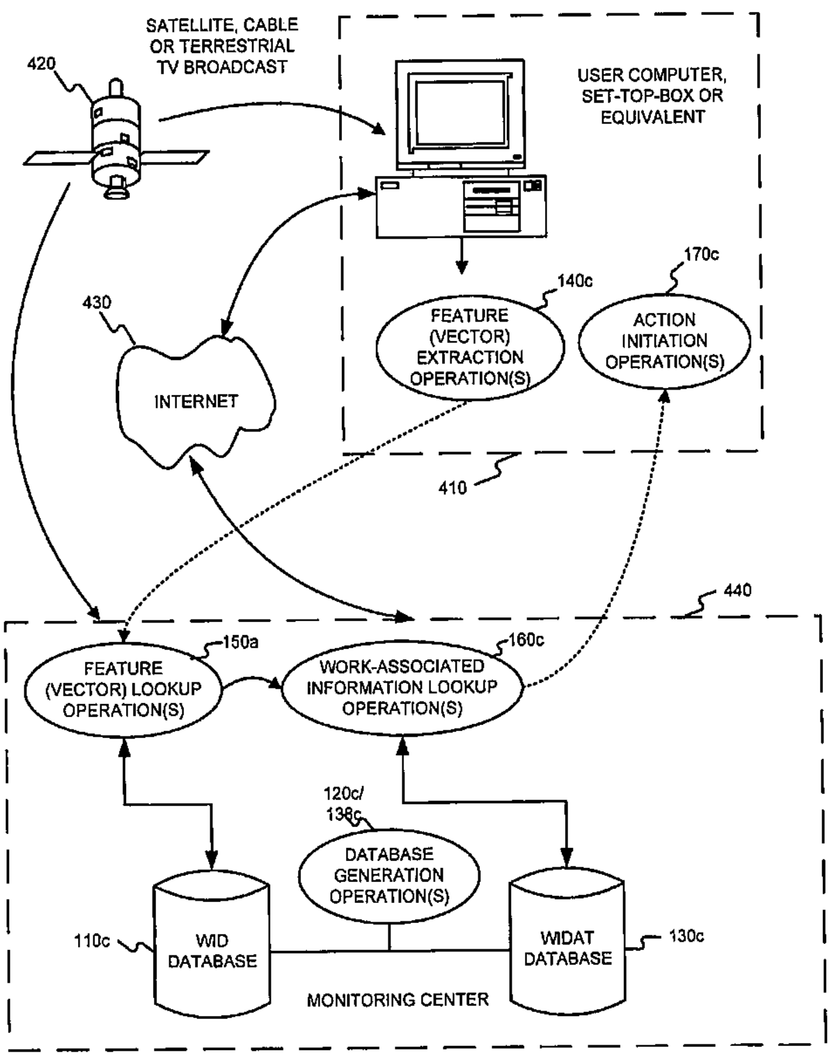

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
420
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
170c
140c
430
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
ACTION
INITIATION
OPERATION(S)
INTERNET
410
440
150a
160c
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
120c/
138c
DATABASE
GENERATION
OPERATION(S)
110c
WID
DATABASE
WIDAT
DATABASE
130c
MONITORING CENTER

FIGURE 4

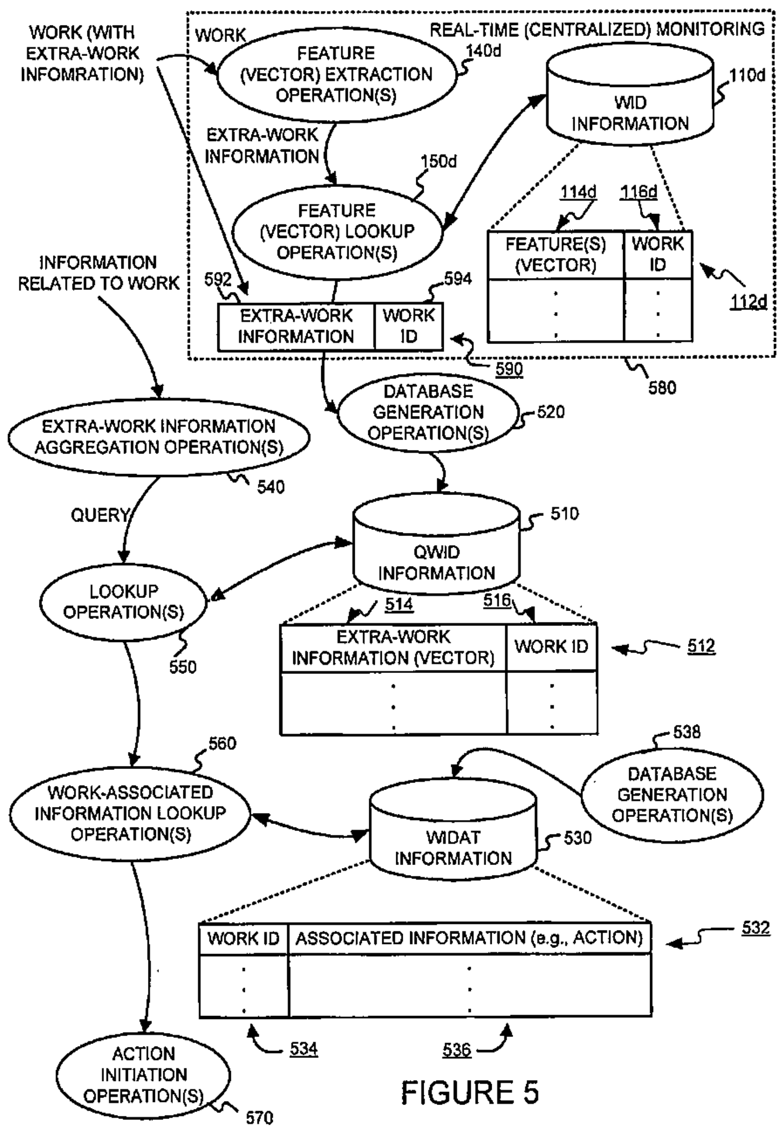

WORK (WITH EXTRA-WORK INFOMRATION)
WORK
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140d
REAL-TIME (CENTRALIZED) MONITORING
WID INFORMATION
110d
EXTRA-WORK INFORMATION
150d
FEATURE (VECTOR) LOOKUP OPERATION(S)
114d
116d
FEATURE(S) (VECTOR)
WORK ID
112d
INFORMATION RELATED TO WORK
592
594
EXTRA-WORK INFORMATION
WORK ID
590
580
DATABASE GENERATION OPERATION(S)
520
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
540
QUERY
510
QWID INFORMATION
LOOKUP OPERATION(S)
550
514
516
EXTRA-WORK INFORMATION (VECTOR)
WORK ID
512
538
DATABASE GENERATION OPERATION(S)
560
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WIDAT INFORMATION
530
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
532
534
536
ACTION INITIATION OPERATION(S)
570

FIGURE 5

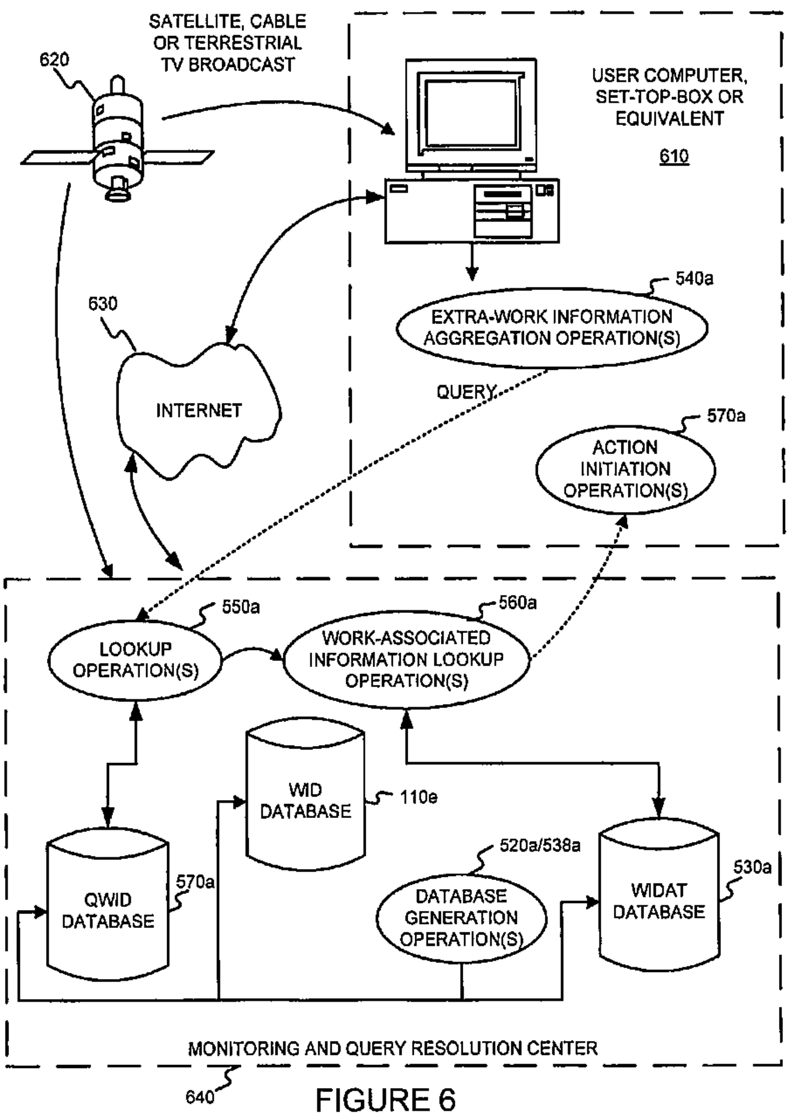

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
620
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
610
540a
EXTRA-WORK INFORMATION
AGGREGATION OPERATION(S)
630
INTERNET
QUERY
570a
ACTION
INITIATION
OPERATION(S)
550a
LOOKUP
OPERATION(S)
560a
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
WID
DATABASE
110e
520a/538a
QWID
DATABASE
570a
DATABASE
GENERATION
OPERATION(S)
WIDAT
DATABASE
530a
MONITORING AND QUERY RESOLUTION CENTER
640

FIGURE 6

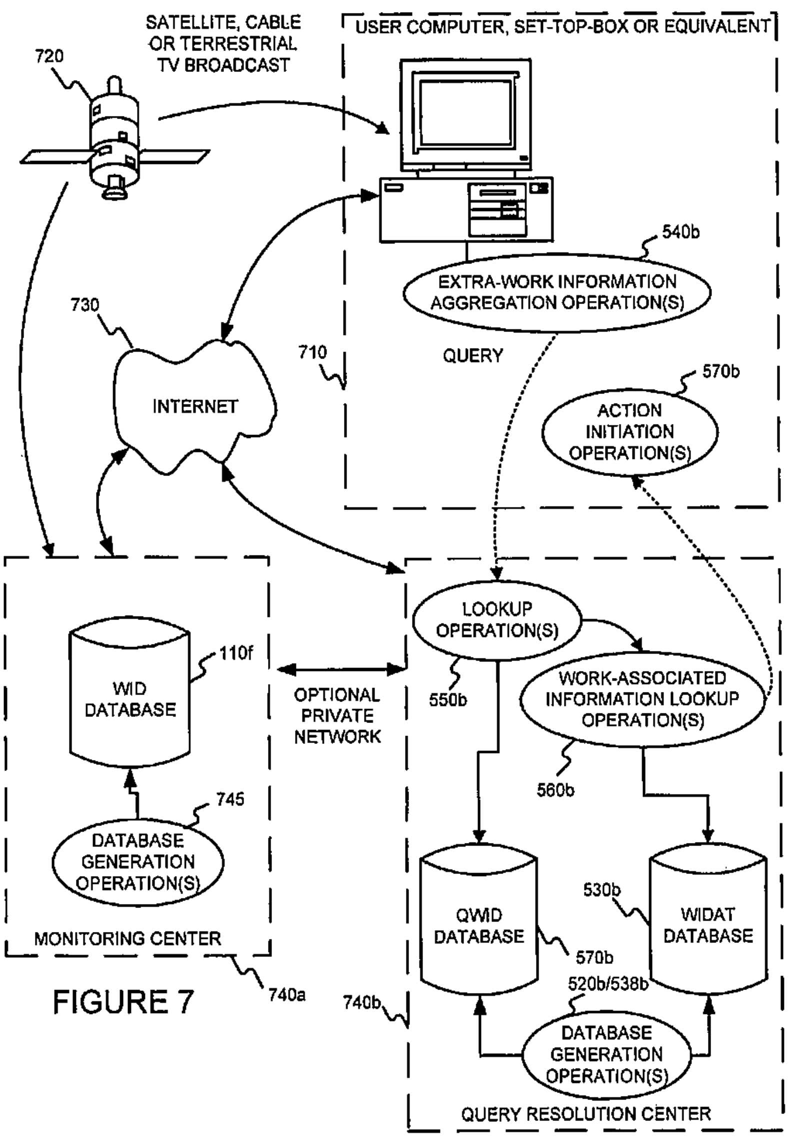

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
720
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
540b
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
730
INTERNET
710
QUERY
570b
ACTION INITIATION OPERATION(S)
LOOKUP OPERATION(S)
110f
WID DATABASE
OPTIONAL PRIVATE NETWORK
550b
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
560b
745
DATABASE GENERATION OPERATION(S)
MONITORING CENTER
740a
QWID DATABASE
570b
530b
WIDAT DATABASE
520b/538b
740b
DATABASE GENERATION OPERATION(S)
QUERY RESOLUTION CENTER

FIGURE 7

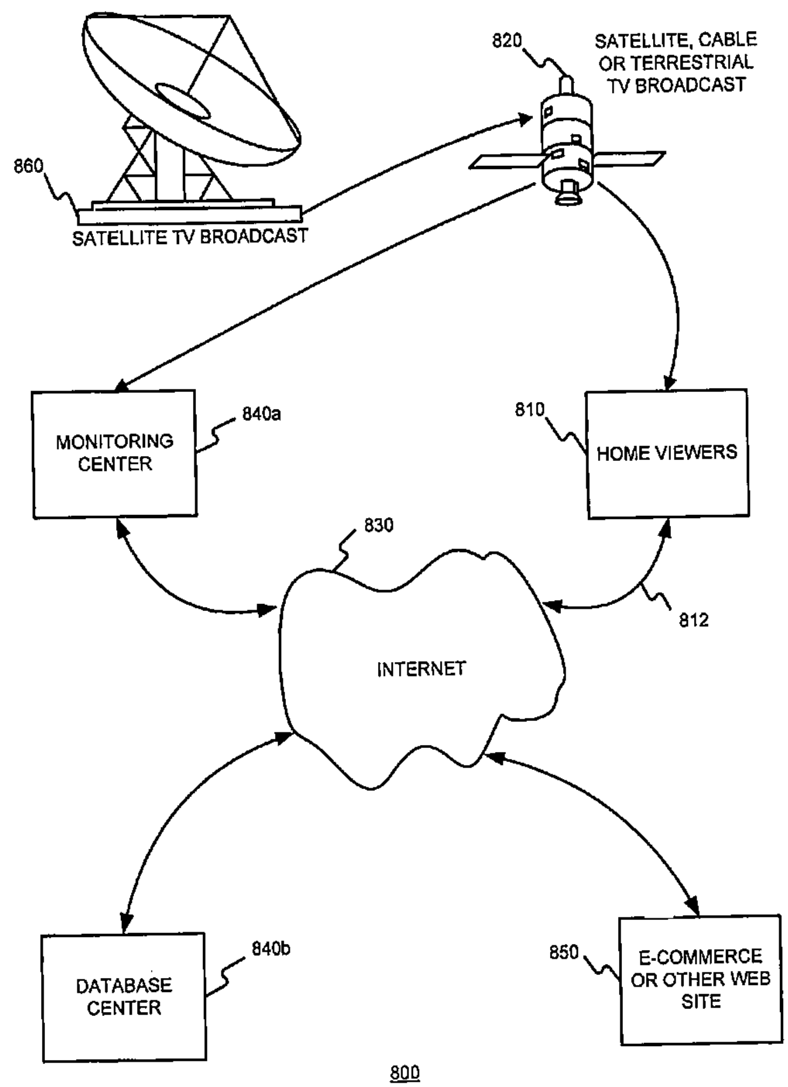

820
SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
860
SATELLITE TV BROADCAST
MONITORING CENTER
840a
810
HOME VIEWERS
830
INTERNET
812
DATABASE CENTER
840b
850
E-COMMERCE OR OTHER WEB SITE
800

FIGURE 8

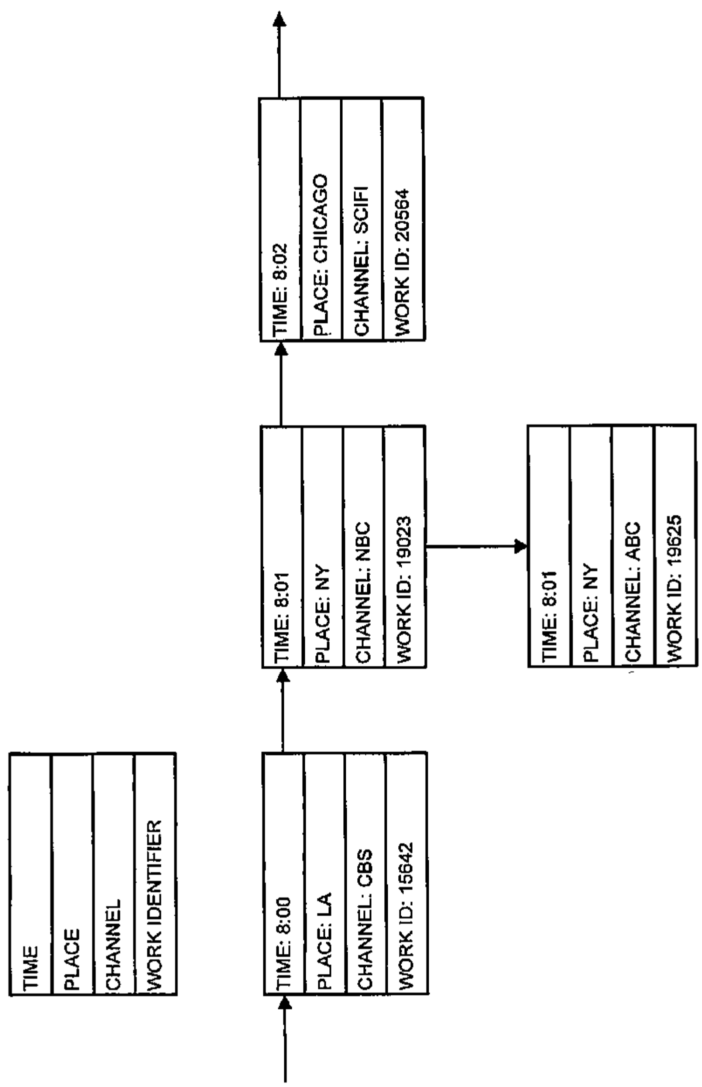

TIME
PLACE
CHANNEL
WORK IDENTIFIER
TIME: 8:00
PLACE: LA
CHANNEL: CBS
WORK ID: 15642
TIME: 8:01
PLACE: NY
CHANNEL: NBC
WORK ID: 19023
TIME: 8:01
PLACE: NY
CHANNEL: ABC
WORK ID: 19625
TIME: 8:02
PLACE: CHICAGO
CHANNEL: SCIFI
WORK ID: 20564

FIGURE 9

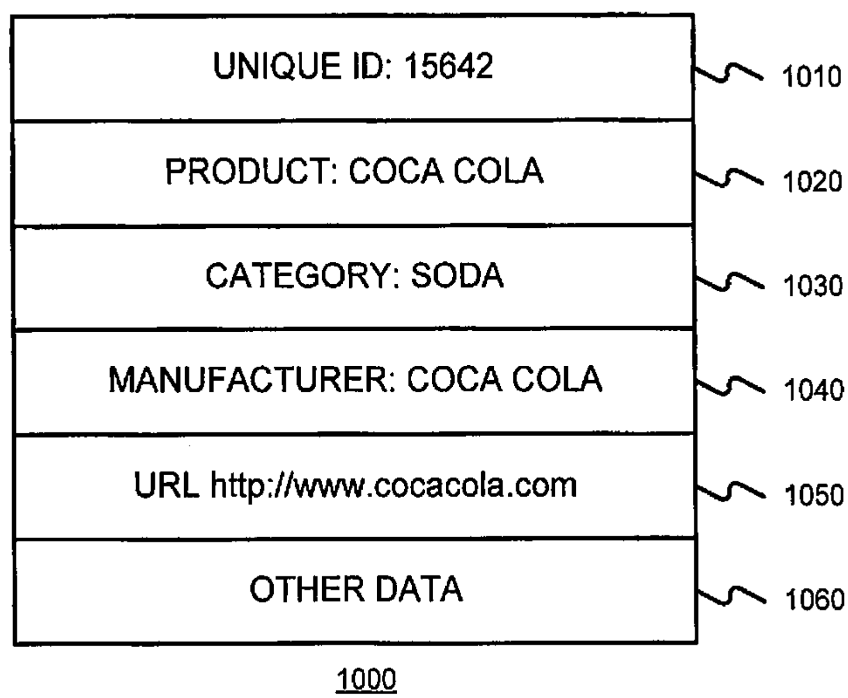

UNIQUE ID: 15642
1010
PRODUCT: COCA COLA
1020
CATEGORY: SODA
1030
MANUFACTURER: COCA COLA
1040
URL http://www.cocacola.com
1050
OTHER DATA
1060
1000

FIGURE 10

IDENTIFYING WORKS, USING A SUB-LINEAR TIME SEARCH, SUCH AS AN APPROXIMATE NEAREST NEIGHBOR SEARCH, FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET

§0. RELATED APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 11/445,928 (incorporated herein by reference), titled "USING FEATURES EXTRACTED FROM AN AUDIO AND/OR VIDEO WORK TO OBTAIN INFORMATION ABOUT THE WORK," filed on Jun. 2, 2006, and listing Ingemar J. Cox as the inventor, which is a continuation-in-part of U.S. patent application Ser. No. 09/950,972 (incorporated herein by reference, issued as U.S. Pat. No. 7,058,223 on Jun. 6, 2006), titled "IDENTIFYING WORKS FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET," filed on Sep. 13, 2001, now U.S. Pat. No. 7,058,223 and listing Ingemar J. Cox as the inventor, which application claims benefit to the filing date of provisional patent application Ser. No. 60/232,618 (incorporated herein by reference), titled "Identifying and linking television, audio, print and other media to the Internet", filed on Sep. 14, 2000 and listing Ingemar J. Cox as the inventor.

§1. BACKGROUND OF THE INVENTION

§1.1 Field of the Invention

The present invention concerns linking traditional media to new interactive media, such as that provided over the Internet for example. In particular, the present invention concerns identifying a work (e.g., content or an advertisement delivered via print media, or via a radio or television broadcast) without the need to modify the work.

§1.2 Related Art

§1.2.1 Opportunities Arising from Linking Works Delivered Via Some Traditional Media Channel or Conduit to a More Interactive System

The rapid adoption of the Internet and associated World Wide Web has recently spurred interest in linking works, delivered via traditional media channels or conduits, to a more interactive system, such as the Internet for example. Basically, such linking can be used to (a) promote commerce, such as e-commerce, and/or (b) enhance interest in the work itself by facilitating audience interaction or participation. Commerce opportunities include, for example, facilitating the placement of direct orders for products, providing product coupons, providing further information related to a product, product placement, etc.

In the context of e-commerce, viewers could request discount vouchers or coupons for viewed products that are redeemable at the point of purchase. E-commerce applications also extend beyond advertisements. It is now common for television shows to include product placements. For example, an actor might drink a Coke rather than a Pepsi brand of soda, actors and actresses might wear designer-labeled clothing such as Calvin Klein, etc. Viewers may wish to purchase similar clothing but may not necessarily be able to identify the designer or the particular style directly from the show. However, with an interactive capability, viewers would be able to discover this and other information by going to an associated Web site. The link to this Web site can be automatically enabled using the invention described herein.

In the context of facilitating audience interaction or participation, there is much interest in the convergence of television and computers. Convergence encompasses a very wide range of capabilities. Although a significant effort is being directed to video-on-demand applications, in which there is a unique video stream for each user of the service, as well as to transmitting video signals over the Internet, there is also interest in enhancing the television viewing experience. To this end, there have been a number of experiments with interactive television in which viewers can participate in a live broadcast. There are a variety of ways in which viewers can participate. For example, during game shows, users can answer the questions and their scores can be tabulated. In recent reality-based programming such as the ABC television game show, "Big Brother", viewers can vote on contestants who must leave the show, and be eliminated from the competition.

§1.2.2 Embedding Work Identifying Code or Signals Within Works

Known techniques of linking works delivered via traditional media channels to a more interactive system typically require some type of code, used to identify the work, to be inserted into the work before it is delivered via such traditional media channels. Some examples of such inserted code include (i) signals inserted into the vertical blanking interval ("VBI") lines of a (e.g., NTSC) television signal, (ii) watermarks embedded into images, (iii) bar codes imposed on images, and (iv) tones embedded into music.

The common technical theme of these proposed implementations is the insertion of visible or invisible signals into the media that can be decoded by a computer. These signals can contain a variety of information. In its most direct form, the signal may directly encode the URL of the associated Web site. However, since the alphanumeric string has variable length and is not a particularly efficient coding, it is more common to encode a unique ID. The computer then accesses a database, which is usually proprietary, and matches the ID with the associated web address. This database can be considered a form of domain name server, similar to those already deployed for network addresses. However, in this case, the domain name server is proprietary and the addresses are unique ID's.

There are two principal advantages to encoding a proprietary identifier into content. First, as previously mentioned, it is a more efficient use of the available bandwidth and second, by directing all traffic to a single Web site that contains the database, a company can maintain control over the technology and gather useful statistics that may then be sold to advertisers and publishers.

As an example of inserting signals into the vertical blanking interval lines of a television signal, RespondTV of San Francisco, Calif. embeds identification information into the vertical blanking interval of the television signal. The VBI is part of the analog video broadcast that is not visible to television viewers. For digital television, it may be possible to encode the information in, for example, the motion picture experts group ("MPEG") header. In the USA, the vertical blanking interval is currently used to transmit close-captioning information as well as other information, while in the UK, the VBI is used to transmit teletext information. Although the close captioning information is guaranteed to be transmitted into the home in America, unfortunately, other information is not. This is because ownership of the vertical blanking interval is disputed by content owners, broadcasters and local television operators.

As an example of embedding watermarks into images, Digimarc of Tualatin, Oreg. embeds watermarks in print media. Invisible watermarks are newer than VBI insertion, and have the advantage of being independent of the method of broadcast. Thus, once the information is embedded, it should remain readable whether the video is transmitted in NTSC, PAL or SECAM analog formats or newer digital formats. It should be more reliable than using the vertical blanking interval in television applications. Unfortunately, however, watermarks still require modification of the broadcast signal which is problematic for a number of economic, logistical, legal (permission to alter the content is needed) and quality control (the content may be degraded by the addition of a watermark) reasons.

As an example of imposing bar codes on images, print advertisers are currently testing a technology that allows an advertisement to be shown to a camera, scanner or bar code reader that is connected to a personal computer ("PC"). The captured image is then analyzed to determine an associated Web site that the PC's browser then accesses. For example, GoCode of Draper, Utah embeds small two-dimensional bar codes for print advertisements. The latter signal is read by inexpensive barcode readers that can be connected to a PC. AirClic of Blue Bell, Pa. provides a combination of barcode and wireless communication to enable wireless shopping through print media. A so-called "CueCat" reads bar codes printed in conjunction with advertisements and articles in Forbes magazine. Similar capabilities are being tested for television and audio media.

Machine-readable bar codes are one example of a visible signal. The advantage of this technology is that it is very mature. However, the fact that the signal is visible is often considered a disadvantage since it may detract from the aesthetic of the work delivered via a traditional media channel or conduit.

As an example of embedding tones into music, Digital Convergence of Dallas, Tex. proposes to embed identification codes into audible music tones broadcast with television signals.

All the foregoing techniques of inserting code into a work can be categorized as active techniques in that they must alter the existing signal, whether it is music, print, television or other media, such that an identification code is also present. There are several disadvantages that active systems share. First, there are aesthetic or fidelity issues associated with bar codes, audible tones and watermarks. More importantly, all media must be processed, before it is delivered to the end user, to contain these active signals. Even if a system is enthusiastically adopted, the logistics involved with inserting bar codes or watermarks into, say every printed advertisement, are formidable.

Further, even if the rate of adoption is very rapid, it nevertheless remains true that during the early deployment of the system, most works will not be tagged. Thus, consumers that are early-adopters will find that most media is not identified. At best, this is frustrating. At worst, the naïve user may conclude that the system is not reliable or does not work at all. This erroneous conclusion might have a very adverse effect on the adoption rate.

Further, not only must there be modification to the production process, but modifications must also be made to the equipment in a user's home. Again, using the example of watermarking of print media, a PC must be fitted with a camera and watermark detection software must be installed. In the case of television, the detection of the identification signal is likely to occur at the set-top-box—this is the equipment provided by the local cable television or satellite broadcasting company. In many cases, this may require modifications to the hardware, which is likely to be prohibitively expensive. For example, the audible tone used by Digital Convergence to recognize television content, must be fed directly into a sound card in a PC. This requires a physical connection between the television and the PC, which may be expensive or at least inconvenient, and a sound card may have to be purchased.

§1.2.3 Unmet Needs

In view of the foregoing disadvantages of inserting an identification code into a work, thereby altering the existing signal, there is a need for techniques of identifying a work without the need of inserting an identification code into a work. Such an identification code can then be used to invoke a work-related action, such as work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§2. SUMMARY OF THE INVENTION

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable programs for linking a media work to an action. Such embodiments might (a) extract features from the media work, (b) determine an identification of the media work based on the features extracted using a sub-linear time search, such as an approximate nearest neighbor search for example, and (c) determine an action based on the identification of the media work determined. In some embodiments consistent with the present invention, the media work is an audio signal. The audio signal might be obtained from a broadcast, or an audio file format. In other embodiments consistent with the present invention, the media work is a video signal. The video signal might be obtained from a broadcast, or a video file format.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§3. BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work.

FIG. 2 is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 3 is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 4 is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 5 is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work.

FIG. 6 is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 8 is a block diagram illustrating an environment in which the present invention may operate.

FIG. 9 is an exemplary data structure in which extra-work information is associated with a work identifier.

FIG. 10 is an exemplary data structure including work-related actions.

§4. DETAILED DESCRIPTION

The present invention may involve novel methods, apparatus and data structures for identifying works without the need of embedding signals therein. Once identified, such information can be used to determine a work-related action. The following description is presented to enable one skilled in the art to make and use the invention, and is provided in the context of particular embodiments and methods. Various modifications to the disclosed embodiments and methods will be apparent to those skilled in the art, and the general principles set forth below may be applied to other embodiments, methods and applications. Thus, the present invention is not intended to be limited to the embodiments and methods shown and the inventors regard their invention as the following disclosed methods, apparatus, data structures and any other patentable subject matter to the extent that they are patentable.

§4.1 FUNCTIONS

The present invention functions to identify a work without the need of inserting an identification code into a work. The present invention may do so by (i) extracting features from the work to define a feature vector, and (ii) comparing the feature vector to feature vectors associated with identified works. Alternatively, or in addition, the present invention may do so by (i) accepting extra-work information, such as the time of a query or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and (ii) use such extra-work information to lookup an identification of the work. In either case, an identification code may be used to identify the work.

The present invention may then function to use such an identification code to initiate a work-related action, such as for work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§4.2 EMBODIMENTS

As just introduced in §4.1 above, the present invention may use intra-work information and/or extra-work information to identify a work. Once identified, such identification can be used to initiate an action, such as an action related to commerce, or facilitating audience participation or interaction. Exemplary embodiments of the present invention, in which work is recognized or identified based on intra-work information, are described in §4.2.1. Then, exemplary embodiments of the present invention, in which work is recognized or identified based on extra-work information, are described in §4.2.2.

§4.2.1 Embodiments in Which Work is Recognized Based on Intra-Work Information

Such as a Feature Vector

Operations related to this embodiment are described in §4.2.1.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.1.2.

§4.2.1.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work. As shown, a work-identification information storage 110 may include a number of items or records 112. Each item or record 112 may associate a feature vector of a work 114 with a, preferably unique, work identifier 116. The work-identification information storage 110 may be generated by a database generation operation(s) 120 which may, in turn, use a feature extraction operation(s) 122 to extract features from a work at a first time ($WORK_{@t1}$), as well as a feature-to-work identification tagging operation(s) 124.

Further, work identifier-action information storage 130 may include a number of items or records 132. Each item or record 132 may associate a, preferably unique, work identifier 134 with associated information 136, such as an action for example. The work identifier-action information storage 130 may be generated by a database generation operation(s) 138 which may, for example, accept manual entries.

As can be appreciated from the foregoing, the work-information storage 110 records 112 and the work identification-action 130 records 132 can be combined into a single record. That is, there need not be two databases. A single database is also possible in which the work identifier, or a feature vector extracted from the work, serves as a key and the associated field contains work-related information, such as a URL for example.

The feature extraction operation(s) 140 can accept a work, such as that being rendered by a user, at a second time ($WORK_{@t2}$), and extract features from that work. The extracted features may be used to define a so-called feature vector.

The extracted features, e.g., as a feature vector, can be used by a feature (vector) lookup operation(s) 150 to search for a matching feature vector 114. If a match, or a match within a predetermined threshold is determined, then the associated work identifier 116 is read.

The read work identifier can then be used by a work-associated information lookup operation(s) 160 to retrieve associated information, such as an action, 136 associated with the work identifier. Such information 136 can then be passed to action initiation operation(s) 170 which can perform some action based on the associated information 136.

§4.2.1.1.1 Exemplary Techniques for Feature Extraction

When the user initiates a request, the specific television or radio broadcast or printed commercial, each of which is referred to as a work, is first passed to the feature extraction operation. The work may be an image, an audio file or some portion of an audio signal or may be one or more frames or fields of a video signal, or a multimedia signal. The purpose of the feature extraction operation is to derive a compact representation of the work that can subsequently be used for the purpose of recognition. In the case of images and video, this feature vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of nxn blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, wavelet and or discrete cosine transforms. It might involve principal component analysis. It might also be a combination of these. For television and audio signals, recognition might also rely on a temporal sequence of feature vectors. The recognition literature contains many different representations. For block-based methods, blocks may be accessed at pseudo-random locations in each frame or might have a specific structure. For audio, common feature vectors are based on Fourier frequency decompositions, but other representations are possible. See, e.g., R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley-Interscience, New York, 1973). See also K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Ed. (Academic Press, New York, 1990). (These references are incorporated herein by reference.)

As previously stated, one object of the vector extraction stage is to obtain a more concise representation of the frame. For example, each video frame is initially composed of 480×720 pixels which is equivalent to 345,600 pixels or 691,200 bytes. In comparison, an exemplary feature vector might only consist of 1 Kbyte of data.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e., horizontal and or vertical translation, or may undergo lossy compression such as by MPEG-2. It is advantageous that these and other processes do not adversely affect the extracted vectors. For still images there has been considerable work on determining image properties that are invariant to affine and other geometric distortions. For example, the use of Radon and Fourier-Mellin transforms have been proposed for robustness against rotation, scale and translation, since these transforms are either invariant or bare a simple relation to the geometric distortions. See, e.g., C. Lin, M. Wu, Y. M. Lui, J. A. Bloom, M. L. Miller, I. J. Cox, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," *IEEE Transactions on Image Processing* (2001). See also, U.S. Pat. Nos. 5,436,653, 5,504,518, 5,582,246, 5,612,729, and 5,621,454. (Each of these references is incorporated herein by reference.)

§4.2.1.1.2 Exemplary Techniques for Database Generation and Maintenance

A number of possibilities exist for generating and maintaining work identification (WID) and identification-action translation (WIDAT) databases. However, in all cases, works of interest are processed to extract a representative feature vector and this feature vector is assigned a unique identifier. This unique identifier is then entered into the work identification (WID) database **110** as well as into the WIDAT database **130** together with all the necessary associated data. This process is referred to as tagging. For example, in the case of an advertisement, the WIDAT database **130** might include the manufacturer (Ford), the product name (Taurus), a product category (automotive) and the URL associated with the Ford Taurus car together with the instruction to translate the query into the associated URL.

The determination of all works of interest and subsequent feature vector extraction and tagging depends on whether content owners are actively collaborating with the entity responsible for creating and maintaining the database. If there is no collaboration, then the database entity must collect all works of interest and process and tag them. While this is a significant effort, it is not overwhelming and is certainly commercially feasible. For example, competitive market research firms routinely tabulate all advertisements appearing in a very wide variety of print media. Newspapers and magazines can be scanned in and software algorithms can be applied to the images to identify likely advertisements. These

possible advertisements can then be compared with advertisements already in the WID database **110**. If there is a match, nothing further need be done. If there is not a match, the image can be sent to a human to determine if the page does indeed contain an advertisement. If so, the operator can instruct the computer to extract the representative feature vector and assign it a unique identifier. Then, the operator can insert this information into the content identification database and as well as update the corresponding WIDAT database **130** with all the necessary associated data. This is continually performed as new magazines and papers include new advertisements to maintain the databases. This is a cost to the database entity. Television and radio broadcasts can also be monitored and, in fact, broadcast monitoring is currently performed by companies such as Nielsen Media research and Competitive Media Reporting. Television and radio broadcasts differ from print media in the real-time nature of the signals and the consequent desire for real-time recognition.

In many cases, advertisers, publishers and broadcasters may wish to collaborate with the database provider. In this case, feature extraction and annotation and/or extra-work information may be performed by the advertiser, advertisement agency, network and/or broadcaster and this information sent to the database provider to update the database. Clearly, this arrangement is preferable from the database provider's perspective. However, it is not essential.

§4.2.1.1.3. Exemplary Techniques for Matching Extracted Features with Database Entries

The extracted feature vector is then passed to a recognition (e.g., feature look-up) operation, during which, the vector is compared to entries of known vectors **114** in a content identification (WID) database **110**. It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search might not be possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used including mutual information, Euclidean distance and Lp-norms. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and false negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match. See, e.g., R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley-Interscience, New York, 1973). See also, U.S. Pat. No. 3,919,474 by W. D. Moon, R. J. Weiner, R. A. Hansen and R. N. Linde, entitled "Broadcast Signal Identification System". (Each of these references is incorporated herein by reference.)

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. Unfortunately, binary search is not possible when taking a noisy signal and trying to find the most similar reference signal. This problem is one of nearest neighbor search in a (high-dimensional) feature space. In previous work, it was not uncommon to perform a linear search of all N entries, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this search can be computationally very expensive.

Other forms of matching include those based on clustering, kd-trees, vantage point trees and excluded middle vantage point forests are possible and will be discussed in more detail later. See, e.g., P. N. Yianilos "Excluded Middle Vantage

Point Forests for nearest Neighbor Search", *Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop*, (Jan. 15, 1999). See also, P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" *SODA* 2000: 361-370. (Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" *SODA* 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point.

If the extracted vector "matches" a known vector in the content identification database, then the work has been identified. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when the vector extracted from a work is not matched to the database even though the work is present in the database. There are several reasons why a work's feature vector may fail to match a feature vector database entry. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will often contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible.

Finally, there is the case where the observed work is not present in the database. In this case, the work can be sent to an operator for identification and insertion in the database.

§4.2.1.1.4 Exemplary Work Based Actions

Assuming that the work is correctly identified, then the identifier can be used to retrieve associated information from the second work identification-action translation (WIDAT) database **130** that contains information **136** associated with the particular work **134**. This information may simply be a corresponding URL address, in which case, the action can be considered to be a form of network address translation. However, in general, any information about the work could be stored therein, together with possible actions to be taken such as initiating an e-commerce transaction. After looking up the work identifier **134** in the WIDAT database **130**, an action is performed on behalf of the user, examples of which has been previously described.

In addition to using the system to allow audience members of a work to connect to associated sites on the Internet, a number of other uses are possible. First, the work identification database **130** allows competitive market research data to be collected (e.g., the action may include logging an event). For example, it is possible to determine how many commercials the Coca Cola Company in the Chicago market aired in the month of June. This information is valuable to competitors such as Pepsi. Thus, any company that developed a system as described above could also expect to generate revenue from competitive market research data that it gathers.

Advertisers often wish to ensure that they receive the advertising time that was purchased. To do so, they often hire commercial verification services to verify that the advertisement or commercial did indeed run at the expected time. To do so, currently deployed systems by Nielsen and CMR embedded active signals in the advertisement prior to the broadcast. These signals are then detected by remote monitoring facilities that then report back to a central system which commercials were positively identified. See for example U.S. Pat. No. 5,629,739 by R. A. Dougherty entitled "Apparatus and method for injecting an ancillary signal into a low energy density portion of a color television frequency spectrum", U.S. Pat. No. 4,025,851 by D. E. Haselwood and C. M. Solar entitled "Automatic monitor for programs broadcast", U.S. Pat. No. 5,243,423 by J. P. DeJean, D. Lu and R. Weissman, entitled "Spread spectrum digital data transmission over TV video", and U.S. Pat. No. 5,450,122 by L. D. Keene entitled "In-station television program encoding and monitoring system and method". (Each of these patents is incorporated herein by reference.) Active systems are usually preferred for advertisement verification because the required recognition accuracy is difficult to achieve with passive systems. The passive monitoring system described herein supports commercial verification.

§4.2.1.2 Exemplary Architectures

Three alternative architectural embodiments in which the first technique may be employed are now described with reference to FIGS. **2**, **3**, and **4**.

FIG. **2** is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work and in which a audience member device **210**, such as a PC for example, receives and renders a work that is consumed by an audience member (user). At some point, the user may wish to perform a work-specific action such as traversing to an associated Web site. Upon initiation of this request, the computer **210** performs the operations **140***a*, **150***a*, **160***a* and **170***a*, such as those shown in FIG. **1**. To reiterate, these operations include a feature extraction operation(s) **140***a*, feature vector lookup or matching operation(s) **150***a* in connection with items or records **112***a* in a work-identification (WID) database **110***a*. If a matching feature vector **114***a* is found, the work-associated information lookup operation(s) **160***a* can use the associated work identifier **116***a* to accessing a work identification-action translation (WIDAT) database **130***a* to retrieve associated information **136***a*, possibly including determining what action should be performed.

As described above, the two databases might be integrated into a single database. However, conceptually, they are described here as separate.

An example illustrating operations that can occur in the first embodiment of FIG. **1**, is now described. Consider a print application, in which say 10,000 advertisements are to be recognized that appear in national newspapers and magazines. If 1 Kbyte is required to store each feature vector then approximately 10 Mbytes of storage will be required for the work identification database **110***a*. Such a size does not represent a serious problem, in either memory or disk space, to present personal computers.

An important issue then becomes recognition rate. While this may be problematic, all the images are two-dimensional—three-dimensional object recognition is not required. Of course, since a low cost camera captures the printed advertisement, there may be a number of geometric distortions that might be introduced together with noise. Nevertheless, the application is sufficiently constrained that adequate recognition rates should be achievable with current state-of-the-art computer vision algorithms. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, Jan. 15, 1999. See also, P. N. Yianilos "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" *SODA* 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. Estimates of the size of the WIDAT database **130***a* depend on what associated information (recall fields **136**) is stored. If, for example, only a URL address is needed, about 20 characters can typically represent most URLs. Thus, the size of the WIDAT database **130***a* would be less than 1 Mbyte.

The configuration just described with reference to FIG. **2** places all of the processing and data on each user's local machine **210**. A number of alternative embodiments, in which some or all of the storage and processing requirements are performed remotely, will be described shortly.

As new works are created and made publicly available, the databases residing on a user's local computer become obsolete. Just as the database provider **240** must continually update the databases in order to remain current, there is also a need to update local databases on devices at audience member premises. This update process can be performed over the Internet **230** in a manner very similar to how software is currently upgraded. It is not necessary to download an entirely new database although this is an option. Rather, only the changes need to be transmitted. During this update process, the user's computer **210** might also transmit information to a central monitoring center **240** informing it of which advertisements the computer user has queried. This type of information is valuable to both advertisers and publishers. Of course, care must be taken to ensure the privacy of individual users of the system. However, it is not necessary to know the identity of individual users for the system to work.

FIG. **3** is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work. Although the WIDAT database can be quite small, as illustrated in the exemplary embodiment described above with respect to FIG. **2**, there is still the problem of keeping this database current. While periodic updates of the local databases may be acceptable, they become unnecessary if the WIDAT database **130***b* is at a remote location **340**. In this arrangement, illustrated in FIG. **3**, after the local computer **310** identifies the work, it sends a query to the remote WIDAT database **130***b*. The query may contain the work identifier. The remote site **340** may then return the associated information **136**. Although the remote WIDAT database **130***b* needs to be updated by the database provider, this can be done very frequently without the need for communicating the updates to the local computers **310**.

The second embodiment is most similar to active systems in which an embedded signal is extracted and decoded and the identifier is used to interrogate a central database. Consequently it has many of the advantages of such systems, while avoiding the need to insert signals into all works. One such advantage, is that the database provider receives real-time information relating to users' access patterns.

The WIDAT database **130***b* might physically reside at more than one location. In such a case, some requests will go to one site, and other requests will go to another. In this way, overloading of a single site by too many users can be avoided. Other load balancing techniques are also applicable.

FIG. **4** is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work. Recall that the WIDAT database may be small relative to that work identification database (WID). As the size of the work recognition (WID) database increases, the foregoing embodiments may become impractical. Consider, for example, a music application in which it is desired to identify 100,000 song titles. If it is again assumed that a 1 Kbyte vector can uniquely represent each song, then on the order of 100 Mbytes is now needed. This size is comparable to large application programs such as Microsoft's Office 2000 suite. Although this still does not represent an inordinate amount of disk space, if this data needs to reside in memory at all times, then very few present machines will have adequate resources. Clearly, at some point, the proposed architectures scales to a point where requirements become impractical. In this case, a further modification to the architecture is possible.

Since the storage and searching of the work-identifier (WID) database require the most computation and storage, it may be more economical to perform these actions remotely. Thus, for example, if a user is playing an MP3 music file and wants to go to a corresponding website, the MP3 file is passed to an operation that determines one or more feature vectors. In the third embodiment, instead of performing the matching locally **410**, the one or more vectors are transmitted to a central site **440** at which is stored the WID and WIDAT databases **110***c* and **130***c* together with sufficiently powerful computers to resolve this request and those of other computer users. This configuration is illustrated in FIG. **4**. Similarly, if a user is playing an MPEG or other video file and wants to initiate a work-related action, the video file is passed to an operation **140***c* that extracts one or more feature vectors. The entire video file need not be processed. Rather, it may be sufficient to process only those frames in the temporal vicinity to the users request, i.e., to process the current frame and or some number of frames before and after the current frame, e.g. perhaps 100 frames in all. The extracted feature vector or feature vectors can then be transmitted to a central site **440** which can resolve the request.

After successfully matching the feature vector, the central site **440** can provide the user with information directly, or can direct the user to another Web site that contains the information the user wants. In cases where the recognition is ambiguous, the central site **440** might return information identifying one of several possible matches and allow the user to select the intended one.

The third embodiment is particularly attractive if the cost of extracting the feature vector is small. In this case, it becomes economical to have feature vector extraction **140***c* in digital set-top-boxes and in video recorders **410**. The latter may be especially useful for the new generation of consumer digital video recorders such as those manufactured by TIVO and Replay TV. These devices already have access to the Internet via a phone line. Thus, when someone watching a recorded movie from television reacts to an advertisement, the video recorder would extract one or more feature vectors and transmit them to a central site **440**. This site **440** would determine if a match existed between the query vector and the database of pre-stored vectors **110***c*. If a match is found, the central server **440** would transmit the associated information, which might include a Web site address or an 800 number for more traditional ordering, back to the audience user device **410**. Of course, a consumer device **410** such as a digital video recorder might also store personal information of the owner to facilitate online e-commerce. Such a device **410** could store the owner's name, address, and credit card information and automatically transmit them to an on-line store to complete a purchase. Very little user interaction other than to authorize the purchase might be needed. This type of purchasing may be very convenient to consumers.

Another advantage of the third embodiment is that it obviates the need to update local databases while, at the same time, the centrally maintained databases can be kept current with very frequent updating.

§4.2.2 Embodiments in which Work is Recognized Based on Extra-Work Information

Operations related to this embodiment are described in §4.2.2.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.2.2.

If the cost of extracting a feature vector is too large, then the cost of deploying any of the embodiments described in §4.2.1 above may be prohibitive. This is particularly likely in very cost sensitive consumer products, including set-top-boxes and next generation digital VCR's. Acknowledging this fact, a different technique, one that is particularly well suited for broadcasted media such as television and radio as well as to content published in magazines and newspapers, is now described. This technique relies on the fact that a work need not be identified by a feature vector extracted from the work (which is an example of "intra-work information"), but can also be identified by when and where it is published or broadcast (which are examples of "extra-work information")

An example serves to illustrate this point. Consider the scenario in which a viewer sees a television commercial and responds to it. The embodiments described in §4.2.1 above required the user device (e.g., a computer or set-top-box) **210/310/410** to extract a feature vector. Such an extracted vector was attempted to be matched to another feature vector(s), either locally, or at a remote site. In the embodiments using a remote site, if the central site is monitoring all television broadcasts, then the user's query does not need to include the feature vector. Instead, the query simply needs to identify the time, geographic location and the station that the viewer is watching. A central site can then determine which advertisement was airing at that moment and, once again, return the associated information. The same is true for radio broadcasts. Moreover, magazines and newspapers can also be handled in this manner. Here the query might include the name of the magazine, the month of publication and the page number.

§4.2.2.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. **5** is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work. As shown, a query work-identification (QWID) information storage **510** may include a number of items or records **512**. Each item or record **512** may associate extra-work information **514**, related to the work, with a, preferably unique, work identifier **516**. The query work-identification (QWID) information storage **510** may be generated by a database generation operation(s) **520**.

Further, work identifier-action information (WIDAT) storage **530** may include a number of items or records **532**. Each item or record **532** may associate a, preferably unique, work identifier **534** with associated information **536**, such as an action for example. The work identifier-action (WIDAT) information storage **530** may be generated by a database generation operation(s) **538** which may, for example, accept manual entries.

As can be appreciated from the foregoing, the query work-information (QWID) storage **510** records **512** and the work identification-action (WIDAT) storage **530** records **532** can be combined into a single record.

The extra-work information aggregation (e.g., query generation) operation(s) **540** can accept a information related to a work, such as the time of a user request or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and generate a query from such extra-work information.

The query including the extra-work information can be used by a lookup operation(s) **550** to search for a "matching" set of information **514**. If a match, or a match within a predetermined threshold is determined, then the associated work identifier **516** is read.

The read work identifier can then be used by a work-associated information lookup operation(s) **560** to retrieve associated information, such as an action, **536** associated with the work identifier. Such information **536** can then be passed to action initiation operation(s) **570** which can perform some action based on the associated information **536**.

If the extra-work information of a work is known (in advance), generating the query work identifier (QWID) information **510** is straight-forward. If this were always the case, an intra-work information-based recognition operation would not be needed. However, very often this is not the case. For example, local television broadcasts typically have discretion to insert local advertising, as well as national advertising. Thus, it often is not possible to know in advance when, on what station, and where a particular advertisement will play.

In such instances, a real-time (e.g., centralized) monitoring facility **580** may be used to (i) extract feature vectors from a work, (ii) determine a work identifier **116** from the extracted features, and (iii) communicate one or more messages **590** in which extra-work information (e.g., time, channel, geographic market) **592** is associated with a work identifier **594**, to operation(s) **520** for generating query work identification (QWID) information **510**.

§4.2.2.1.1 Exemplary Extra-Work Information

In the context of national broadcasts, geographic information may be needed to distinguish between, for example, the ABC television broadcast in Los Angeles and that in New York. While both locations broadcast ABC's programming, this programming airs at different times on the East and West coasts of America. More importantly, the local network affiliates that air ABC's shows have discretion to sell local advertising as well as a responsibility to broadcast the national commercials that ABC sells. In short, the works broadcast by ABC in Los Angeles can be different from that in other geographic locations. Geographic information is therefore useful to distinguish between the different television markets. In some circumstances, geographic information may not be necessary, especially in parts of the world with highly regulated and centralized broadcasting in which there are not regional differences.

§4.2.2.1.2 Exemplary Techniques for Generating Databases

FIG. **5** illustrates a third database **510** referred to as the query to work identification (QWID) database. This database

510 maps the query (e.g., in the form of time, location and channel information) into a unique ID that identifies the perceived work. The QWID **510** and WIDAT **530** databases might not be separate, but for clarity will be considered so. After retrieving the unique work identifier **512** from the QWID database **510**, the identifier can be used to access the WIDAT database **530**. This is discussed in more detail later.

As introduced above, although it appears that this architecture does not require a recognition facility, such a facility may be needed. The feature extraction operation(s) **140***d*, as well as the work identification operation(s) **150***d* and other databases **110***d*, may be moved to one or more remote sites **580**.

Although TV Guide and other companies provide detailed information regarding what will be broadcast when, these scheduling guides do not have any information regarding what advertisements will air when. In many cases, this information is unknown until a day or so before the broadcast. Even then, the time slots that a broadcaster sells to an advertiser only provide a time range, e.g. 12 pm to 3 pm. Thus it is unlikely that all commercials and aired programming can be determined from TV schedules and other sources prior to transmission. Further, occasionally programming schedules are altered unexpectedly due to live broadcasts that overrun their time slots. This is common in sports events and awards shows. Another example of interrupts to scheduled programming occurs when a particularly important news event occurs.

During transmission, it may therefore be necessary for a central site **580** to determine what work is being broadcast and to update its and/or other's database **520** accordingly based on the work identified **594** and relevant extra-work information **592**. There are a variety of ways that this can be accomplished.

First, it may be economically feasible to manually monitor all television stations that are of interest, and manually update the database with information regarding the work being monitored. In fact, Nielsen used such procedures in the early 1960's for the company to tabulate competitive market data. More than one person can be employed to watch the same channel in order to reduce the error rate. It should be noted that the recent ruling by the FCC that satellite broadcasters such as DirecTV, DishTV and EchoStar can carry local stations significantly reduces the cost of monitoring many geographic markets. Currently, DirecTV, for example, carries the four main local stations in each of the 35 largest markets. Thus, these 4x35=140 channels can all be monitored from a single site **580**. This site would be provided with satellite receivers to obtain the television channels.

Unfortunately, however, humans are error prone and the monitoring of many different stations from many different geographic locations can be expensive. In order to automate the recognition process, a central site **580** could employ a computer-based system to perform automatic recognition. Because the recognition is centralized, only one or a few sites are needed. This is in comparison with the first architecture we described in which a complete recognition system was required in every user's home or premise. This centralization makes it more economic to employ more expensive computers, perhaps even special purpose hardware, and more sophisticated software algorithms. When video frames or clips cannot be identified or are considered ambiguous, this video can be quickly passed to human viewers to identify. Further, it should be possible for the automated recognition system to use additional information such as television schedules, time of day, etc in order to improve its recognition rate.

§4.2.2.1.2 Exemplary Techniques for Generating Queries Based on Extra-Work Information

At the audience member (user) premises, all that is needed is for the device to send a query to a database-server with information that includes extra-work information, such as geographic location, time and channel. Usually, this extra-work information would be transmitted in real-time, while the work (e.g., an advertisement) is being broadcast. However, this is not necessary. If the television does not have access to the Internet, and most TV's do not yet, then an audience member (user) may simply remember or record which channel he or she was viewing at what time. In fact, the user device could store this information for later retrieval by the user. At a convenient later time, the user might access the Internet using a home PC. At this time, he or she can query the database by entering this extra-work information (e.g., together with geographic information) into an application program or a web browser plug-in.

Another possibility is allowing an audience member (user), at the time he or she is consuming (e.g., viewing, reading, listening to, etc.) the work, to enter query information into a handheld personal digital assistant ("PDA") such as a Palm Pilot, so as not to forget it. This information can then be manually transferred to a device connected to a network, or the information can be transferred automatically using, for example, infrared communications or via a physical link such as a cradle. Recently, PDAs also have some wireless networking capabilities built in, and thus might support direct access to the information desired. Further, software is available that allows a Palm Pilot or other PDA to function as a TV remote control device. As such, the PDA already knows the time of day and channel being viewed. It also probably knows the location of the audience member, since most PDA users include their own name and address in the PDA's phonebook and identify it as their own. Thus, with one or a few clicks, an audience member PDA user could bookmark the television content he or she is viewing. If the PDA is networked, then the PDA can, itself, retrieve the associated information immediately. Otherwise, the PDA can transfer this bookmarked data to a networked device, which can then provide access to the central database.

§4.2.2.2 Exemplary Architectures

FIG. **6** is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work. As shown, an extra-work information aggregation operation **540***a* may be effected on a device **610**, such as a PC, at the audience member (user) premises. The various databases **510***a*, **530***a*, and **110***e*, as well as the database generation operation(s) **520***a*/**538***a*, the lookup operation(s) **550***a* and the work-associated information lookup operation(s) **560***a* may be provided at one or more centralized monitoring and query resolution centers **640**.

FIG. **7** is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work. This fifth embodiment is similar to the fourth embodiment illustrated in FIG. **6** but here, the monitoring center **740***a* and query resolution center **740***b* are separate.

These embodiments have many advantages for television and radio broadcasters who desire to provide Internet links or other action. First, the audience member (user) equipment, whether it is a computer, set-top-box, television, radio, remote control, personal digital assistant (pda), cell phone or other device, does not need to perform any processing of the received signal. As such, there is almost no cost involved to equipment manufacturers.

These last embodiments have some similarity with services such as those provided by the companies Real Names of Redwood City, Calif., America Online ("AOL") and especially iTag from Xenote. The popular press has reported on the difficulties associated with assigning domain names. The simplest of these problems is that almost all the one-word names in the ".com" category have been used. Consequently, domain names can often be difficult to remember. To alleviate this problem, RealNames and AOL provide alternative, proprietary name spaces (AOL calls these keywords). For a fee, a company may register a name with these companies. Thus, rather than type the URL http://www.bell-labs.com, the simple keyword "bell" might be sufficient to access the same Web site. These capabilities are convenient to users. However, these systems are very different from the fourth and fifth embodiments described. First, and foremost, these systems are not designed to identify content. Rather, they are simply alternative network address translation systems based on easily remembered mnemonics which are sold to interested companies. As such, the user is still expected to type in an address, but this address is easier to remember than the equivalent URL. In contrast, while a user may manually enter the information describing the work, the preferred embodiment is for the computer, set-top-box or other device to automatically generate this information. Further, the mapping of keywords to network addresses is an arbitrary mapping maintained by AOL or Real Names. For example, the keyword "bell" might just as reasonably point to the Web site for Philadelphia's Liberty Bell as to Lucent's Bell Labs. In contrast, the query used in the fourth and fifth embodiments is designed to contain all the necessary data to identify the work, e.g. the time, place and television channel during which the work was broadcast. There is nothing arbitrary about this mapping. It should also be pointed out that the proposed system is dynamic—the same work, e.g. a commercial, potentially has an infinite number of addresses depending on when and where it is broadcast. If an advertisement airs 100,000 unique times, then there are 100,000 different queries that uniquely identify it. Moreover, the exemplary query includes naturally occurring information such as time, place, channel or page number. This is not the case for AOL or RealNames, which typically assigns one or more static keywords to the address of a Web site.

Xenote's iTag system is designed to identify radio broadcasts and uses a query similar to that which may be used in the fourth and fifth embodiments, i.e. time and station information. However, the work identification information is not dynamically constructed but is instead based on detailed program scheduling that radio stations must provide it. As such, it suffers from potential errors in scheduling and requires the detailed cooperation of broadcasters. While the fourth and fifth embodiments might choose to use program scheduling information and other ancillary information to aid in the recognition process, they do not exclusively rely on this. The concept of resolving a site name by recognizing the content is absent from the above systems.

§4.2.3 Exemplary Apparatus for Audience Member (User) Premise Device

While personal computers may be the primary computational device at a user's location, it is not essential to use a PC. This is especially true of the embodiments depicted in FIGS. 6 and 7, which do not require the content, e.g. video signal, to be processed. Instead, only a unique set of identification parameters such as time, location and channel are provided to identify the perceived Work. Many forms of devices can therefore take advantage of this configuration.

As previously noted, personal digital assistants (PDAs) can be used to record the identification information. This information can then be transferred to a device with a network communication such as a PC. However, increasingly, PDAs will already have wireless network communication capabilities built-in, as with the Palm VII PDA. These devices will allow immediate communication with the query resolution center and all information will be downloaded to them or they can participate in facilitating an e-commerce transaction. Similarly, wireless telephones are increasingly offering web-enabled capabilities. Consequently, wireless phones could be programmed to act as a user interface.

New devices can also be envisaged, including a universal remote control for home entertainment systems with a LCD or other graphical display and a network connection. This connection may be wireless or the remote control might have a phone jack that allows it to be plugged directly into an existing phone line. As home networks begin to be deployed, such devices can be expected to communicate via an inexpensive interface to the home network and from there to access the Internet.

In many homes, it is not uncommon for a computer and television to be used simultaneously, perhaps in the same room. A person watching television could install a web browser plug-in or applet that would ask the user to identify his location and the station being watched. Then, periodically, every 20 seconds for example, the plug-in would update a list of web addresses that are relevant to the television programs being watched, including the commercials. The audience member would then simply click on the web address of interest to obtain further information. This has the advantage that the viewer does not have to guess the relevant address associated with a commercial and, in fact, can be directed to a more specialized address, such as www.fordvehicles.com/ibv/tausrus2kflash/flash.html, rather than the generic www.ford.com site. Of course, this applet or plug-in could also provide the database entity with information regarding what is being accessed from where and at what time. This information, as noted earlier, is valuable to advertisers and broadcasters. For PC's that have infra-red communication capabilities, it is straightforward to either control the home entertainment center from the PC or for the PC to decode the signals from a conventional remote control. Thus, as a user changes channels, the PC is able to automatically track the channel changes.

Recording devices such as analog VCR's and newer digital recording devices can also be exploited in the embodiments depicted in FIGS. 6 and 7, especially if device also record the channel and time information for the recorded content. When a user initiates a query, the recorded time and channel, rather than the current time and channel, then form part of the identification information.

Digital set-top-boxes are also expected to exploit the capabilities described herein. In particular, such devices will have two-way communication capabilities and may even include cable modem capabilities of course, the two-way communication need not be over a television cable. For example, satellite set-top-boxes provide up-link communications via a telephone connection. Clearly, such devices provide a convenient location to enable the services described herein. Moreover, such services can be provided as part of the OpenCable and DOCSIS (data over cable service interface specification) initiatives.

§4.2.4 Information Retrieval Using Features Extracted from Audio and/or Video Works

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable program for providing information about an audio file or (a video file) played on a device. Such embodiments might (a) extract features from the audio (or video) file, (b) communicate the features to a database, and (c) receive the information about the audio (or video) file from the database. In some embodiments consistent with the present invention, the act of extracting the features is performed by a microprocessor of the device, and/or a digital signal processor of the device. The received information might be rendered on an output (e.g., a monitor, a speaker, etc.) of the device. The received information might be stored (e.g., persistently) locally on the device. The information might be stored on a disk, or non-volatile memory.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§4.3 OPERATIONAL EXAMPLES

An example illustrating operations of an exemplary embodiment of the present invention, that uses intra-work information to identify the work, is provided in §4.3.1. Then, an example illustrating operations of an exemplary embodiment of the present invention, that uses extra-work information to identify the work, is provided in §4.3.2.

§4.3.1 Operational Example where Intra-Work Information is Used to Identify the Work

A generic system for monitoring television commercials is now described. Obviously, the basic ideas extend beyond this specific application.

The process of recognition usually begins by recognizing the start of a commercial. This can be accomplished by looking for black video frames before and after a commercial. If a number of black frames are detected and subsequently a similar number are detected 30 seconds later, then there is a good chance that a commercial has aired and that others will follow. It is also well known than the average sound volume during commercials is higher than that for television shows and this too can be used as an indicator of a commercial. Other methods can also be used. The need to recognize the beginning of a commercial is not essential. However, without this stage, all television programming must be assumed to be commercials. As such, all video frames must be analyzed. The advantage of determining the presence of a commercial is that less video content must be processed. Since the percentage of advertising time is relatively small, this can lead to considerable savings. For example, commercials can be buffered and then subsequently processed while the television show is being broadcast. This reduces the real-time requirements of a system at the expense of buffering, which requires memory or disk space. Of course, for the applications envisioned herein, a real-time response to a user requires real-time processing.

Once it is determined that an advertisement is being broadcast, it is necessary to analyze the video frames. Typically, a compact representation of each frame is extracted. This vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of nxn blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, Fourier-Mellin, wavelet and or discrete cosine transforms. It might involve principal component analysis or any combination thereof. The recognition literature contains many different representations. For block-based methods, the nxn blocks may be located at pseudo-random locations in each frame or might have a specific structure, e.g. a complete tiling of the frame. The feature vector might then be composed of the pixels in each block or some property of each block, e.g. the average intensity or a Fourier or other decomposition of the block. The object of the vector extraction stage is to obtain a more concise representation of the frame. Each frame is initially composed of 480×720 pixels which is equivalent to 345,600 bytes, assuming one byte per pixel. In comparison, the feature vector might only consist of 1 Kbyte of data. For example, if each frame is completely tiled with 16×16 blocks, then the number of blocks per frame is 345,600/256=1350. If the average intensity of each block constitutes the feature vector, then the feature vector consists of 1350 bytes, assuming 8-bit precision for the average intensity values. Alternatively, 100 16×16 blocks can be pseudo-randomly located on each frame of the video. For each of these 100 blocks, the first 10 DCT coefficients can be determined. The feature vector then consists of the 100×10=1000 DCT coefficients. Many other variations are also possible. In many media applications, the content possesses strong temporal and spatial correlations. If necessary, these correlations can be eliminated or substantially reduced by pre-processing the content with a whitening filter.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e. horizontal and or vertical translation, or may undergo lossy compression such as MPEG-2. It is advantageous, though not essential, that these and other processes do not adversely affect the extracted vectors.

Each frame's feature vector is then compared with a database of known feature vectors. These known vectors have previously been entered into a content recognition database together with a unique identifier. If a frame's vector matches a known vector, then the commercial is recognized. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when a frame's vector is not matched to the database even though the advertisement is present in the database. There are several reasons why a frame's feature vector may fail to match. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible. Finally, there is the case where the observed commercial is not yet present in the database. In this case, it is necessary to store the commercial and pass it (e.g., to a person) for identification and subsequent entry in the database.

It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search is often not possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used, including clustering techniques. See, e.g., the Duda and

Google Ex. 1001

Google Ex. 1020

Hart reference. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match.

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. However, in current advertisement monitoring applications there is no discussion of efficient search methods. Thus, a linear search of all N entries may be performed, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this can be computationally expensive. Consider a situation in which one out of 100,000 possible commercials is to be identified. Each 30-second commercial consists of 900 video frames. If all 900 frames are stored in the database, then N=90,000,000. Even if only every 10^{th} video frame is stored in the database, its size is still nine million. While databases of this size are now common, they rely of efficient search to access entries, i.e., they do not perform a linear search. A binary search of a 90,000,000-item database requires less than 20 comparisons. In contrast, a linear search will require an average of 45,000,000!

With 9 million entries, if each vector is 1 Kbyte, then the storage requirement is 9 Gigabytes. Disk drives with this capacity are extremely cheap at this time. However, if the database must reside in memory due to real-time requirements, then this still represents a substantial memory requirement by today's standards. One reason that the data may need to be stored in memory is because of the real-time requirements of the database. If 10 channels are being simultaneously monitored within each of 50 geographic areas, then there will be 15,000 queries per second to the content recognition database, assuming each and every frame is analyzed. This query rate is low. However, if a linear search is performed then 675 billion comparisons per second will be required. This is an extremely high computational rate by today's standards. Even if only key frames are analyzed, this is unlikely to reduce the computational rate by more than an order of magnitude.

If an advertisement is not recognized, then typically, the remote monitoring system will compress the video and transmit it back to a central office. Here, the clip is identified and added to the database and the remote recognition sites are subsequently updated. Identification and annotation may be performed manually. However, automatic annotation is also possible using optical character recognition software on each frame of video, speech recognition software, close captioning information and other information sources. As these methods improve in accuracy, it is expected that they will replace manual identification and annotation.

The recognition system described can be considered to be a form of nearest neighbor search in a high dimensional feature space. This problem has been very well studied and is known to be very difficult as the dimensionality of the vectors increases. A number of possible data structures are applicable including kd-trees and vantage point trees. These data structures and associated search algorithms organize a N-point dataset (N=90,000,000 in out previous example) so that sub-linear time searches can be performed on average. However, worst-case search times can be considerably longer. Recently, Yianilos proposed an excluded middle vantage point forest for nearest neighbor search. See, e.g., the Yianilos reference. This data structure guarantees sub-linear worst-case search times, but where the search is now for a nearest neighbor within a fixed radius, τ. The fixed radius search means that if the database contains a vector that is within τ of the query, then there is a match. Otherwise, no match is found. In contrast, traditional vantage point trees will always return a nearest neighbor, even if the distance between the neighbor and the query is very large. In these cases, if the distance between the query and the nearest neighbor exceeds a threshold, then they are considered not to match. This is precisely what the excluded middle vantage point forest implicitly does.

Using an excluded middle vantage point forest, will allow accurate real-time recognition of 100,000 broadcasted advertisements. This entails constructing an excluded middle vantage point forest based on feature vectors extracted from say 90,000,000 frames of video. Of course, using some form of pre-filtering that eliminates a large number of redundant frames or frames that are not considered to be good unique identifiers can reduce this number. One such pre-filter would be to only examine the I-frames used when applying MPEG compression. However, this is unlikely to reduce the work identification database (WID) size by more than one order of magnitude. Assuming 10 channels are monitored in each of 50 geographic regions, then the query rate is 15,000=10×50×30 queries per second.

§4.3.2 Operational Example where Extra-Work Information is Used to Identify the Work

FIG. **8** depicts a satellite television broadcast system **800**, though cable and traditional broadcast modes are also applicable. Block **810** represents audience members (users) watching a TV channel in their home, which also has a connection **812** to the Internet **820**. Other networks are also possible. The satellite broadcasts are also being monitored by one or more television monitoring centers **840***a*. These centers **840***a* may monitor all or a subset of the television channels being broadcast. They are not restricted to monitoring satellite TV broadcasts but may also monitor cable and traditional terrestrial broadcasts. The primary purpose of these monitoring centers **840***a* is to identify the works being broadcasted. Of particular interest are television advertisements. However, other works, or portions thereof, may also be identified. Each time a new segment of a work is identified, the monitoring system or systems **840***a* update one or more database centers **840***b*, informing them of the time, place, channel and identity of the identified segment. The segment may be a complete thirty second commercial or, more likely, updates will occur more frequently, perhaps at a rate of 1 update per second per channel per geographic location. The database center **840***b* updates its database so that queries can be efficiently responded to in sub-linear time.

The database centers **840***b* can use traditional database technology. In general, the query search initiated by an audience member is not a nearest neighbor search but can be a classical textual search procedure such as a binary search. The nearest neighbor search is appropriate for the monitoring sub-system **840***a*. The database centers **840***b* are continually updated as each new advertisement, television show or portion thereof is recognized. Standard updating algorithms can be used. However, random new entries to the database are unlikely. Rather, each new entry, or set of entries, denotes a new time segment that is later than all previously inserted items. As such, each new entry can be appended to the end of the database while still maintaining an ordered data structure that is amenable to binary and other efficient search techniques. If two entries have the same time in their time field, items can be sorted based on secondary fields such as the channel and geographic location, as depicted in FIG. **9**. Since the number of such entries will be relatively small compared with the entire database, it may be sufficient to simply create a linear linked list of such entries, as depicted in FIG. 9. Of course, the size of the database is constantly increasing. As such, it may become necessary to have several levels of storage and caching. Given the envisaged application, most user queries will be for recent entries. Thus, the database may keep the last hours worth of entries in memory. If there is one entry per second for each of 100 channels in 100 geographic locations, this would correspond to 3600×100×100=36,000,000 entries which is easily accommodated in main memory. Entries that are older than one hour may be stored on disk and entries older than one week may be archived (e.g., backed up on tape) for example. The entries to this database can include time, location and channel information together with a unique identifier that is provided by the monitoring system. Of course, additional fields for each entry are also possible.

When a user query is received, the time, channel and geographic information are used to retrieve the corresponding unique identifier that is then used to access a second database that contains information associated with the identified work.

An entry **1000** in this second database is depicted in FIG. **10**, which shows that associated with the unique identifier **1010**, the name of a product **1020**, a product category **1030**, the manufacturer **1040** and the commercial's associated web site **1050**. Many other data fields **1060** are also possible. Such additional fields may include fields that indicate what action should be taken on behalf of the requesting user. Example actions include simply redirecting a request to an associated Web site, or initiating an e-commerce transaction or providing an associated telephone number that may be automatically dialed if the querying device is a cell phone or displaying additional information to the user. This database is likely to be updated much less frequently, perhaps only as often as once or twice a day, as batches of new advertisements are added to the system. Alternatively, it might be updated as each new advertisement is added to the system.

An audience member (user) **810** watching a television commercial for example may react to the advertisement by initiating a query to the database center **840***b*. The device whereby the user initiates the query might be a television or set-top-box remote control, or a computer or a wireless PDA or a (WAP-enabled) cell phone or a specialized device. Typically, the query will occur during the airing of the commercial or a shortly thereafter. However, the time between the broadcasting of the advertisement and the time of the associated query is not critical and can, in some instances be much longer. For example, the audience member might bookmark the query information in a device such as a PDA or a specialized device similar to those developed by Xenote for their Itag radio linking. Later, the audience member may transmit the query to the database center **840***b*. This might happen hours or even days later.

The query contains information that the database center **840***b* uses to identify the work being viewed. This information might include the time and place where the audience member was, together with the channel being viewed. Other identifying information is also possible. The query may also contain additional information that may be used to facilitate the user's transaction and will include the return address of the user. For example, if the user is intending to order a pizza after seeing a Pizza Hut advertisement, the query may also contain personal information including his or her identity, street address and credit card information.

When the database center **840***b* receives a query, data in the query is used to identify the work and associated information. A number of possible actions are possible at this point. First, the database center **840***b* may simply function as a form of proxy server, mapping the audience member's initial query into a web address associated with the advertisement. In this case, the audience member will be sent to the corresponding Web site. The database center **840***b* may also send additional data included in the initial query to this Web site **850** in order to facilitate an e-commerce transaction between the audience member and the advertiser. In some cases, this transaction will not be direct, but may be indirect via a dealer or third party application service provider. Thus, for example, though an advertisement by Ford Motor Company may air nationally, viewers may be directed to different Web sites for Ford dealerships depending on both the audience member's and the dealerships' geographic locations. In other cases, advertisers may have contracted with the database center **840***b* to provide e-commerce capabilities. This latter arrangement has the potential to reduce the amount of traffic directed over the public Internet, restricting it, instead to a private network associated with the owner of the database center.

If the audience member (user) is not watching live television but is instead watching a taped and therefore time-shifted copy, then additional processes are needed. For the new generation of digital video recorders, irrespective of the recording media (tape or disk), it is likely to be very easy to include information identifying the location of the recorder, as well as the time and channel recorded. Location information can be provided to the recorder during the setup and installation process, for example. Digital video recorders, such as those currently manufactured by TIVO of Alviso, Calif. or Replay TV of Santa Clara, Calif. have a network connection via telephone, which can then send the query of an audience member to the database center **840***b* using the recorded rather than the current information.

In cases where query information has not been recorded, it is still possible to initiate a successful query. However, in this case, it may be necessary to extract the feature vector from the work of interest and send this information to the monitoring center **840***a* where the feature vector can be identified. This form of query is computationally more expensive but the relative number of such queries compared to those sent to the database centers **840***b* is expected to be small. It should also be noted that the physical separation of the monitoring and database centers, depicted in FIGS. **6** and **7**, is not crucial to operation of the system and simply serves to more clearly separate the different functionality present in the overall system configuration.

Although the implementation architectures described above focus on the television media, it is apparent that the present invention is applicable to audio, print and other media.

§4.4 CONCLUSIONS

None of the embodiments of the invention require modification to the work or content, i.e., no active signal is embedded. Consequently, there is no change to the production processes. More importantly, from a user perspective, deployment of this system need not suffer from poor initial coverage. Provided the database is sufficiently comprehensive, early adopters will have comprehensive coverage immediately. Thus, there is less risk that the consumer will perceive that the initial performance of the deployed system is poor. Further, the present invention permits statistics to be gathered that measure users' responses to content. This information is expected to be very useful to advertisers and publishers and broadcasters.

What is claimed is:

1. A computer-implemented method comprising:

a) receiving, by a computer system including at least one computer, features that were extracted from a media work by a client device;

b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and

c) transmitting, by the computer system, information about the identified media work to the client device.

2. The computer-implemented method of claim 1 wherein the media work is an audio work,

wherein the features extracted from the work comprise at least one selected from a group consisting of (A) a frequency decomposition of a signal of the audio work, (B) information samples of the audio work, (C) average intensities of sampled windows of the audio work, and (D) information from frequencies of the audio work, and

wherein the audio work is one of (A) a broadcast, (B) a digital file, or (C) an MP3 file.

3. The computer-implemented method of claim 1 wherein the information about the identified media work transmitted to the client device includes at least one of (A) a title, or (B) an author.

4. The computer-implemented method of claim 1 further comprising performing an action including at least one of promoting commerce or enhancing interest in the work.

5. Apparatus comprising:

a) at least one processor; and

b) at least one storage device storing processor-executable instructions which, when executed by the at least one processor, perform a method of

1) receiving features that were extracted from a media work by a client device,

2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and

3) transmitting information about the identified media work to the client device.

6. The apparatus of claim 5 wherein the media work is an audio work,

wherein the features extracted from the work comprise at least one selected from a group consisting of (A) a frequency decomposition of a signal of the audio work, (B) information samples of the audio work, (C) average intensities of sampled windows of the audio work, and (D) information from frequencies of the audio work, and

wherein the audio work is one of (A) a broadcast, (B) a digital file, or (C) an MP3 file.

7. The apparatus of claim 5 wherein the information about the identified media work transmitted to the client device includes at least one of (A) a title, or (B) an author.

8. The apparatus of claim 5 wherein the method further includes performing an action including at least one of promoting commerce or enhancing interest in the work.

9. A computer-implemented method comprising:

a) receiving, by a computer system including at least one computer, features what were extracted from media work by a client device;

b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform an approximate nearest neighbor search of extracted features of identified media works; and

c) transmitting, by the computer system, information about the identified media work to the client device.

10. The method of claim 9 wherein the media work is an audio work,

wherein the features extracted from the work comprise at least one selected from a group consisting of (A) a frequency decomposition of a signal of the audio work, (B) information samples of the audio work, (C) average intensities of sampled windows of the audio work, and (D) information from frequencies of the audio work, and

wherein the audio work one of (A) a broadcast, (B) a digital file, or (C) an MP3 file.

11. The method of claim 9 wherein the information about the identified media work transmitted to the client device includes at least one of (A) a title, or (B) an author.

12. The method of claim 9 further comprising performing an action including at least one of promoting commerce or enhancing interest in the work.

13. Apparatus comprising:

a) at least one processor; and

b) at least one storage device storing processor-executable instructions which, when executed by the at least one processor, perform a method of

1) receiving features what were extracted from a media work by a client device,

2) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform an approximate nearest neighbor search of extracted features of identified media works, and

3) transmitting information about the identified media work to the client device.

14. The apparatus of claim 13 wherein the media work is an audio work,

wherein the features extracted from the work comprise at least one selected from a group consisting of (A) a frequency decomposition of a signal of the audio work, (B) information samples of the audio work, (C) average intensities of sampled windows of the audio work, and (D) information from frequencies of the audio work, and

wherein the audio work is one of (A) a broadcast, (8) a digital file, or (C) an MP3 file.

15. The apparatus of claim 13 information about the identified media work transmitted to the client device includes at least one of (A) a title, or (B) an author.

16. The apparatus of claim 13 wherein the method further includes performing an action including at least one of promoting commerce or enhancing interest in the work.

17. The computer-implemented method of claim 1 wherein the media work is a video signal.

18. The computer-implemented method of claim 17 wherein the video signal is obtained from at least one of (A) a broadcast or (B) a video file format.

19. The computer-implemented method of claim 9 wherein the media work is a video signal.

20. The computer-implemented method of claim 19 wherein the video signal is obtained from at least one of (A) a broadcast or B) a video file format.

21. The computer-implemented method of claim 1 wherein at least one of the acts of receiving or transmitting is performed via a direct communication between the client device and the computer system.

22. The computer-implemented method of claim 1 wherein at least one of the acts of receiving or transmitting is per-

formed via an indirect communication between the client device and the computer system.

23. The computer-implemented method of claim 9 wherein at least one of the acts of receiving or transmitting is performed via a direct communication between the client device and the computer system.

24. The computer-implemented method of claim 9 wherein at least one of the acts of receiving or transmitting is performed via an indirect communication between the client device and the computer system.

25. A computer-implemented method comprising:

a) obtaining, by a computer system including at least one computer, media work extracted features that were extracted from a media work, the media work uploaded from a client device;

b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a nonexhaustive search of reference extracted features of reference media works to identify a near neighbor; and

c) determining, by the computer system, an action based on the determined identification of the media work.

26. The method of claim 25, wherein the action comprises providing to and/or displaying, at another client device, additional information in association with the media work.

27. The method of claim 26, wherein the additional information is an advertisement.

28. The method of claim 25, wherein the action comprises providing a coupon.

29. The method of claim 25, wherein the action comprises providing a link to a Web site.

30. The method of claim 25, wherein the action comprises initiating an e-commerce transaction.

31. The method of claim 25, wherein the action comprises initiating a telephone call.

32. The method of claim 25, wherein the action comprises logging an event relating to competitive market research data.

33. A computer-implemented method comprising:

a) obtaining, by a computer system including at least one computer, media work extracted features that were extracted from a media work, the media work uploaded from a client device;

b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a sublinear approximate nearest neighbor search of reference extracted features of reference identified media works; and

c) determining, by the computer system, an action based on the determined identification of the media work.

34. The method of claim 33, wherein the action comprises providing to and/or displaying, at another client device, additional information in association with the media work.

35. The method of claim 34, wherein the additional information is an advertisement.

36. The method of claim 33, wherein the action comprises providing a coupon.

37. The method of claim 33, wherein the action comprises providing a link to a Website.

38. The method of claim 33, wherein the action comprises initiating an e-commerce transaction.

39. The method of claim 33, wherein the action comprises initiating a telephone call.

40. The method of claim 33, wherein the action comprises logging an event relating to competitive market research data.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,205,237 B2
APPLICATION NO. : 11/977202
DATED : June 19, 2012
INVENTOR(S) : Cox

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims

Column 25, line 64, claim 9, "what" should read --that--.

Column 26, line 14, claim 10, "work one" should read --work is one--.

Column 26, line 27, claim 13, "what" should read --that--.

Column 26, line 44, claim 14, "(8)" should read --(B)--.

Column 26, line 46, claim 15, "claim 13 information" should read --claim 13 wherein information--.

Column 26, line 61, claim 20, "B)" should read --(B)--.

Signed and Sealed this
Sixth Day of May, 2014

Michelle K. Lee

Michelle K. Lee
Deputy Director of the United States Patent and Trademark Office

Google Ex. 1001

US008640179B1

(12) **United States Patent**
Cox

(10) **Patent No.:** **US 8,640,179 B1**
(45) **Date of Patent:** ***Jan. 28, 2014**

(54) **METHOD FOR USING EXTRACTED FEATURES FROM AN ELECTRONIC WORK**

(75) Inventor: **Ingemar J. Cox**, London (GB)

(73) Assignee: **Network-1 Security Solutions, Inc.**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 8 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/338,079**

(22) Filed: **Dec. 27, 2011**

Related U.S. Application Data

(63) Continuation of application No. 11/977,202, filed on Oct. 23, 2007, now Pat. No. 8,205,237, which is a continuation of application No. 11/445,928, filed on Jun. 2, 2006, now Pat. No. 8,010,988, which is a continuation-in-part of application No. 09/950,972, filed on Sep. 13, 2001, now Pat. No. 7,058,223.

(60) Provisional application No. 60/232,618, filed on Sep. 14, 2000.

(51) **Int. Cl.**
H04N 7/173 (2011.01)

(52) **U.S. Cl.**
USPC **725/115**; 725/110; 725/114; 725/116

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,919,479 A	11/1975	Moon et al.
4,230,990 A	10/1980	Lert, Jr. et al.
4,450,531 A	5/1984	Kenyon et al.
4,454,594 A	6/1984	Heffron et al.
4,495,526 A	1/1985	Baranoff-Rossine
4,499,601 A	2/1985	Matthews
4,511,917 A	4/1985	Kohler et al.
4,547,804 A	10/1985	Greenberg
4,634,966 A	1/1987	Nakatani et al.
4,639,779 A	1/1987	Greenberg
4,677,455 A	6/1987	Okajima

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1 354 276 B1	12/2007
EP	1354276 B1	12/2007

(Continued)

OTHER PUBLICATIONS

Peter N. Yianlos, Excluded Middle Vantage Point Forest for Nearest Neighbor Search, Aug. 1, 1999, pp. 1-12.*
Martin Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996.

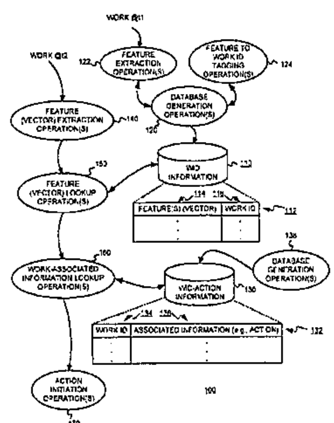
(Continued)

Primary Examiner — Cai Chen
(74) *Attorney, Agent, or Firm* — Amster, Rothstein & Ebenstein LLP

(57) **ABSTRACT**

A computer-implemented method including the steps of maintaining, by a computer system including at least one computer, a database in which is stored first data related to identification of one or more works and second data related to information corresponding to each of the one or more works as identified by the first data. Extracted features of a work to be identified are obtained. The work is identified by comparing the extracted features of the work with the first data in the database using a non-exhaustive neighbor search. The information corresponding to the identified work is determined based on the second data in the database. The determined information is associated with the identified work.

37 Claims, 10 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

4,677,466 A 6/1987 Lert, Jr. et al.
4,682,370 A 7/1987 Matthews
4,697,209 A 9/1987 Kiewit et al.
4,739,398 A 4/1988 Thomas et al.
4,776,017 A 10/1988 Fujimoto
4,805,020 A 2/1989 Greenberg
4,843,526 A 6/1989 Price, III
4,843,562 A 6/1989 Kenyon et al.
4,918,730 A 4/1990 Schulze
5,210,820 A 5/1993 Kenyon
5,283,819 A 2/1994 Glick et al.
5,437,050 A 7/1995 Lamb et al.
5,465,353 A 11/1995 Hull et al.
5,481,294 A 1/1996 Thomas et al.
5,504,518 A 4/1996 Ellis et al.
5,550,735 A 8/1996 Slade et al.
5,581,658 A 12/1996 O'Hagan et al.
5,594,934 A 1/1997 Lu et al.
5,629,739 A 5/1997 Dougherty
5,634,012 A 5/1997 Stefik et al.
5,638,443 A 6/1997 Stefik et al.
5,692,213 A 11/1997 Goldberg et al.
5,701,452 A 12/1997 Siefert
5,701,542 A 12/1997 Sasayama
5,706,364 A 1/1998 Kopec et al.
5,724,605 A 3/1998 Wissner
5,745,900 A 4/1998 Burrows
5,748,783 A 5/1998 Rhoads
5,768,426 A 6/1998 Rhoads
5,798,785 A 8/1998 Hendricks et al.
5,822,436 A 10/1998 Rhoads
5,832,119 A 11/1998 Rhoads
5,841,978 A 11/1998 Rhoads
5,850,490 A 12/1998 Johnson
5,862,260 A 1/1999 Rhoads
5,892,536 A 4/1999 Logan et al.
5,918,223 A 6/1999 Blum et al.
5,953,415 A 9/1999 Nielsen
5,963,966 A 10/1999 Mitchell et al.
5,973,723 A 10/1999 DeLuca
5,978,791 A 11/1999 Farber et al.
5,983,171 A 11/1999 Yokoyama et al.
5,983,176 A 11/1999 Hoffert et al.
6,006,256 A 12/1999 Zdepski et al.
6,009,410 A 12/1999 LeMole et al.
6,011,758 A 1/2000 Dockes et al.
6,023,693 A 2/2000 Masuoka et al.
6,026,439 A 2/2000 Chowdhury et al.
6,044,402 A 3/2000 Jacobson et al.
6,052,693 A 4/2000 Smith et al.
6,057,872 A 5/2000 Candelore
6,061,056 A * 5/2000 Menard et al. 715/704
6,067,369 A 5/2000 Kamei
6,088,455 A 7/2000 Logan et al.
6,088,707 A 7/2000 Bates et al.
6,096,961 A 8/2000 Bruti et al.
6,118,450 A 9/2000 Proehl et al.
6,119,124 A 9/2000 Broder et al.
6,121,530 A 9/2000 Sonoda
6,154,737 A 11/2000 Inaba et al.
6,169,986 B1 1/2001 Bowman et al.
6,173,406 B1 1/2001 Wang et al.
6,195,693 B1 2/2001 Berry et al.
6,229,922 B1 5/2001 Sasakawa et al.
6,233,682 B1 5/2001 Fritsch
6,236,758 B1 5/2001 Sodagar et al.
6,240,409 B1 5/2001 Aiken
6,243,725 B1 6/2001 Hempleman et al.
6,247,133 B1 6/2001 Palage et al.
6,253,193 B1 6/2001 Ginter et al.
6,263,348 B1 7/2001 Kathrow et al.
6,269,275 B1 7/2001 Slade
6,279,010 B1 8/2001 Anderson
6,317,885 B1 11/2001 Fries
6,330,593 B1 12/2001 Roberts et al.
6,345,256 B1 2/2002 Milsted et al.
6,349,296 B1 2/2002 Broder et al.
6,360,215 B1 3/2002 Judd et al.
6,363,377 B1 3/2002 Kravets et al.
6,374,225 B1 4/2002 Hejna, Jr.
6,374,260 B1 4/2002 Hoffert et al.
6,381,601 B1 4/2002 Fujiwara et al.
6,385,596 B1 5/2002 Wiser et al.
6,407,680 B1 6/2002 Lai et al.
6,408,128 B1 6/2002 Abecassis
6,415,280 B1 7/2002 Farber et al.
6,418,421 B1 7/2002 Hurtado et al.
6,438,556 B1 8/2002 Malik et al.
6,446,068 B1 9/2002 Kortge
6,449,226 B1 9/2002 Kumagai
6,452,874 B1 9/2002 Otsuka et al.
6,453,252 B1 9/2002 Laroche
6,460,050 B1 10/2002 Pace et al.
6,477,704 B1 11/2002 Cremia
6,490,279 B1 12/2002 Chen et al.
6,496,802 B1 12/2002 Van Zoest et al.
6,496,857 B1 12/2002 Dustin et al.
6,505,160 B1 1/2003 Levy et al.
6,542,869 B1 4/2003 Foote
6,550,001 B1 4/2003 Corwin et al.
6,550,011 B1 4/2003 Sims, III
6,552,254 B2 4/2003 Hasegawa et al.
6,577,746 B1 6/2003 Evans et al.
6,591,245 B1 7/2003 Klug
6,598,228 B2 7/2003 Hejna, Jr.
6,609,105 B2 8/2003 Story et al.
6,654,757 B1 11/2003 Stern
6,665,661 B1 12/2003 Crow et al.
6,675,174 B1 1/2004 Bolle et al.
6,693,236 B1 2/2004 Gould et al.
6,766,523 B2 7/2004 Herley
6,774,926 B1 8/2004 Ellis et al.
6,810,388 B1 10/2004 Sato
6,833,865 B1 12/2004 Fuller et al.
6,834,308 B1 * 12/2004 Ikezoye et al. 709/231
6,871,200 B2 3/2005 MacQueen et al.
6,871,231 B2 3/2005 Morris
6,873,982 B1 3/2005 Bates et al.
6,928,423 B1 8/2005 Yamanaka
6,928,442 B2 8/2005 Farber et al.
6,931,451 B1 8/2005 Logan et al.
6,937,766 B1 8/2005 Wilf et al.
6,941,275 B1 9/2005 Swierczek
6,944,632 B2 9/2005 Stern
6,968,337 B2 11/2005 Wold
6,978,419 B1 12/2005 Kantrowitz
6,978,461 B2 12/2005 Shapiro et al.
6,983,371 B1 1/2006 Hurtado et al.
6,990,453 B2 1/2006 Wang et al.
6,999,111 B2 2/2006 McIntyre et al.
7,013,301 B2 3/2006 Holm et al.
7,020,635 B2 3/2006 Hamilton et al.
7,043,473 B1 5/2006 Rassool et al.
7,058,223 B2 6/2006 Cox
7,065,709 B2 6/2006 Ellis et al.
7,092,953 B1 8/2006 Haynes
7,103,906 B1 9/2006 Katz et al.
7,106,904 B2 9/2006 Shima
7,155,449 B2 12/2006 Pingel et al.
7,158,929 B2 1/2007 Wouters et al.
7,168,083 B2 1/2007 Kalker et al.
7,171,016 B1 1/2007 Rhoads
7,174,293 B2 2/2007 Kenyon et al.
7,184,100 B1 2/2007 Wilf et al.
7,191,190 B2 3/2007 Debique et al.
7,225,455 B2 5/2007 Bennington et al.
7,243,153 B2 7/2007 McIntyre et al.
7,272,788 B2 9/2007 Anderson et al.
7,302,574 B2 11/2007 Conwell et al.
7,308,413 B1 12/2007 Tota et al.
7,346,472 B1 3/2008 Moskowitz et al.
7,363,278 B2 4/2008 Schmelzer et al.
7,366,718 B1 4/2008 Pugh et al.
7,366,787 B2 4/2008 Salas et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,369,677	B2	5/2008	Petrovic et al.
7,370,017	B1	5/2008	Lindeman et al.
7,421,723	B2	9/2008	Harkness et al.
7,423,771	B2	9/2008	Ohata et al.
7,477,739	B2	1/2009	Haitsma et al.
7,483,958	B1	1/2009	Elabbady et al.
7,493,643	B2	2/2009	Ellis
7,500,007	B2	3/2009	Ikezoye et al.
7,523,312	B2	4/2009	Kalker et al.
7,529,659	B2	5/2009	Wold
7,562,012	B1	7/2009	Wold et al.
7,562,392	B1	7/2009	Rhoads et al.
7,565,327	B2	7/2009	Schmelzer
7,587,728	B2	9/2009	Wheeler et al.
7,595,914	B2	9/2009	Haining
7,624,337	B2	11/2009	Sull et al.
7,647,604	B2	1/2010	Ramaswamy
7,650,616	B2	1/2010	Lee
7,660,700	B2	2/2010	Moskowitz et al.
7,707,088	B2	4/2010	Schmelzer
7,711,652	B2	5/2010	Schmelzer
7,738,704	B2	6/2010	Lienhart et al.
7,757,248	B2	7/2010	Harkness et al.
7,783,489	B2	8/2010	Kenyon et al.
7,797,249	B2	9/2010	Schmelzer et al.
7,853,664	B1	12/2010	Wang et al.
7,877,438	B2	1/2011	Schrempp et al.
7,917,645	B2	3/2011	Ikezoye et al.
7,949,494	B2	5/2011	Moskowitz et al.
8,006,314	B2	8/2011	Wold
8,082,150	B2	12/2011	Wold
8,086,445	B2	12/2011	Wold et al.
8,090,605	B2	1/2012	Tota et al.
8,094,949	B1	1/2012	Rhoads
8,112,776	B2	2/2012	Schein et al.
8,171,509	B1	5/2012	Girouard et al.
8,214,175	B2	7/2012	Moskowitz et al.
8,340,994	B2	12/2012	Tota et al.
2001/0001160	A1 *	5/2001	Shoff et al. 725/51
2001/0003818	A1	6/2001	Pingel et al.
2001/0047298	A1	11/2001	Moore et al.
2001/0049625	A1	12/2001	Mowry
2002/0023020	A1	2/2002	Kenyon et al.
2002/0026369	A1	2/2002	Miller et al.
2002/0032698	A1	3/2002	Cox
2002/0038296	A1	3/2002	Margolus et al.
2002/0056123	A1	5/2002	Liwerant et al.
2002/0059610	A1	5/2002	Ellis
2002/0082731	A1	6/2002	Pitman et al.
2002/0083005	A1	6/2002	Lowenstein et al.
2002/0087885	A1	7/2002	Peled et al.
2002/0120925	A1	8/2002	Logan
2002/0133499	A1	9/2002	Ward et al.
2002/0150164	A1	10/2002	Felts et al.
2002/0156760	A1	10/2002	Lawrence et al.
2002/0178276	A1	11/2002	McCartney et al.
2002/0186887	A1	12/2002	Rhoads
2003/0028489	A1	2/2003	Williamson
2003/0037010	A1	2/2003	Schmelzer
2003/0061490	A1	3/2003	Abajian
2003/0093790	A1	5/2003	Logan et al.
2003/0095660	A1	5/2003	Lee et al.
2003/0101144	A1	5/2003	Moreno
2003/0106017	A1	6/2003	Kanchirayappa et al.
2003/0146940	A1	8/2003	Ellis et al.
2003/0202660	A1	10/2003	Zhou et al.
2003/0233930	A1	12/2003	Ozick
2004/0003398	A1	1/2004	Donian et al.
2004/0010602	A1	1/2004	Van Vleck et al.
2004/0015608	A1	1/2004	Ellis et al.
2004/0025174	A1	2/2004	Cerrato
2004/0163106	A1	8/2004	Schrempp et al.
2004/0170335	A1	9/2004	Pearlman et al.
2004/0199387	A1 *	10/2004	Wang et al. 704/243
2004/0221118	A1	11/2004	Slater et al.
2004/0243540	A1	12/2004	Moskowitz et al.
2005/0044189	A1	2/2005	Ikezoye et al.
2005/0080846	A1	4/2005	McCleskey et al.
2005/0102515	A1	5/2005	Jaworski et al.
2005/0154892	A1	7/2005	Mihcak et al.
2005/0160363	A1	7/2005	Bhogal et al.
2005/0193016	A1	9/2005	Seet et al.
2005/0213826	A1	9/2005	Neogi
2005/0246752	A1	11/2005	Liwerant et al.
2005/0289065	A1	12/2005	Weare
2006/0031870	A1	2/2006	Jarman et al.
2006/0080356	A1	4/2006	Burges et al.
2006/0085816	A1	4/2006	Funk et al.
2006/0101069	A1	5/2006	Bell et al.
2006/0110137	A1	5/2006	Tsuda et al.
2006/0187358	A1	8/2006	Lienhart et al.
2006/0195859	A1	8/2006	Konig et al.
2006/0195860	A1	8/2006	Eldering et al.
2006/0206462	A1	9/2006	Barber
2006/0212927	A1	9/2006	Riku et al.
2006/0271947	A1	11/2006	Lienhart et al.
2007/0041667	A1	2/2007	Cox
2007/0071330	A1	3/2007	Oostveen et al.
2007/0083510	A1	4/2007	McArdle
2007/0101360	A1	5/2007	Gutta et al.
2007/0118375	A1	5/2007	Kenyon et al.
2007/0124698	A1	5/2007	Majumder
2007/0130580	A1	6/2007	Covell et al.
2007/0180537	A1	8/2007	He et al.
2007/0203911	A1	8/2007	Chiu
2007/0282472	A1	12/2007	Seidman
2007/0288518	A1	12/2007	Crigler et al.
2008/0052783	A1	2/2008	Levy
2008/0091684	A1	4/2008	Ellis et al.
2008/0250241	A1	10/2008	Ginter et al.
2009/0052784	A1	2/2009	Covell et al.

FOREIGN PATENT DOCUMENTS

EP	1 485 815	B1	7/2009
GB	2369203	A	5/2002
JP	2003-242281		8/2003
WO	94/06084	A1	3/1994
WO	99/50778	A1	10/1999
WO	WO0122730	A1	3/2001
WO	WO 2002/011033	A1	2/2002
WO	WO 2002/103968	A1	12/2002

OTHER PUBLICATIONS

Yossi Rubner et al., "Adaptive Color Image Embeddings for Database Navigation*," Proceedings of the 1998 IEEE Asian Conference on Computer Vision.

Roger Weber et al., "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High Dimensional Spaces," Proceedings of 24th VLDB Conference, 1998.

P. Yianilos, "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces," Proceedings of the ACM-SIAM Symposium on Discrete algorithms, 1993, pp. 311-321.

Peter N. Yianlos, Excluded Middle Vantage Point Forests for Nearest Neighbor Search, Jul. 20, 1998, pp. 1-12.

Peter N. Yianlos "Locally Lifting the Curse of Dimensionality for Nearest Neighbor Search" SODA 2000, pp. 361-370.

L. Baum et al., "A Maximation Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains," The Annals of Mathematical Statistics, vol. 41, No. 1, pp. 164-171 (1970).

A. P. Dempster et al., "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, Series B (Methodological), vol. 39, Issue 1, pp. 1-38 (1977).

D. Reynolds et al., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," IEEE Transactions on Speech and Audio Processing, vol. 3, No. 1, pp. 72-83 (Jan. 1995).

D. Bouktache, "A fast Algorithm for the nearest neighbor classifier," IEEE Transactions on Pattern Analysis and Machine Intelligence, Mar. 1997, pp. 277-282.

Nene et al., "A simple algorithm for nearest neighbor search in high dimensions," IEEE Transactions on Pattern Analysis and Machine Intelligence, Sep. 1997, pp. 989-1003.

(56) **References Cited**

OTHER PUBLICATIONS

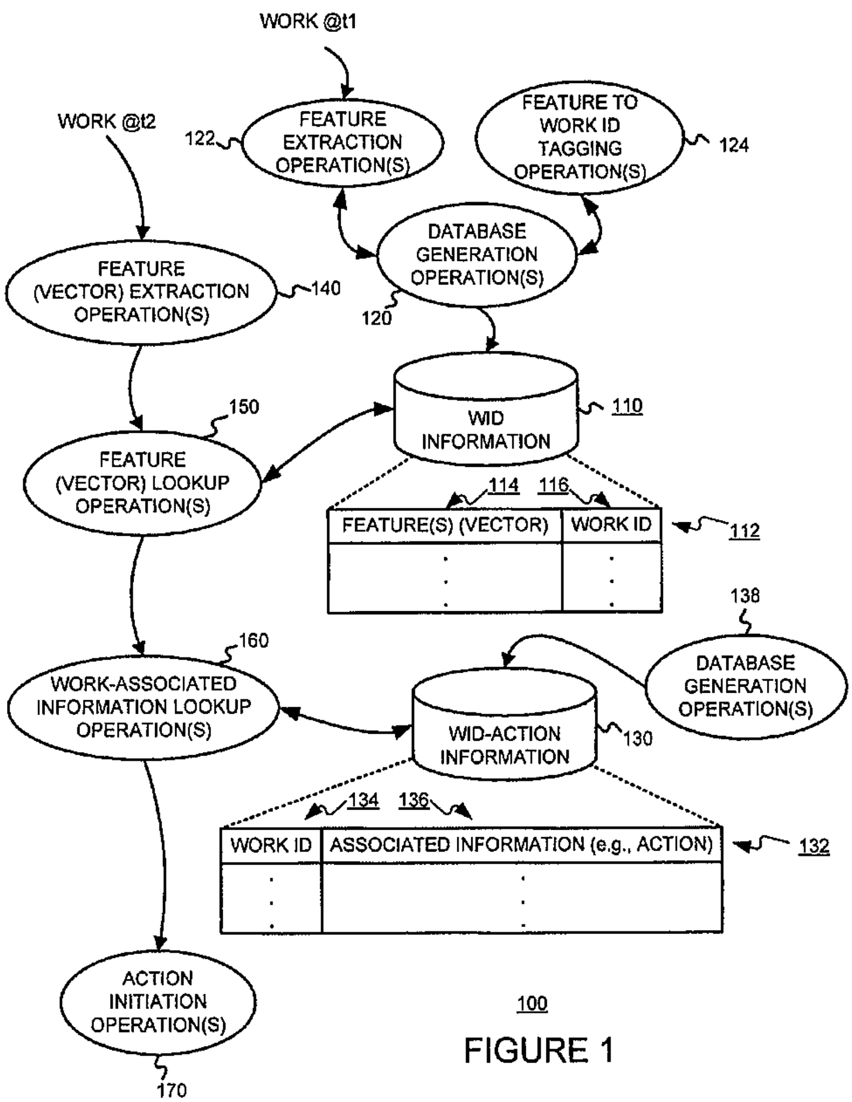
Sunil Arya et al., "Approximate Nearest Neighbor Queries in Fixed Dimensions," Proceedings of the 4th annual ACM-SIAM Symposium on Discrete algorithms, 1993, pp. 271-280.
K. Fukunaga et al., A branch and bound algorithm for computing k-nearest neighbors, IEEE Trans. Comput., C24:750-753, Jul. 1975.
Charles D. Feustel et al., "The nearest neighbor problem in an abstract metric space," Pattern Recognition Letters, pp. 125-128, Dec. 1982.
Dennis Shasha et al., "New Techniques for Best-Match Retrieval," ACM Transactions on Information Systems, 8(2); 140{158, Apr. 1990.
J. Uhlmann, "Satisfying general proximity / similarity queries with metric trees", Information Processing Letters, 40 (4): 175{9, Nov. 1991.
Sergey Brin, "Near Neighbor Search in Large Metric Spaces," Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995.
Daniel P. Huttenlocher, et al., "Comparing Images Using the Hausdorff Distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, No. 9, pp. 850-863, Sep. 1993.
Thomas Seidl et al., "Optimal Multi-Step K-Nearest Neighbor Search," Proceedings of ACM SIGMOD International Conference of Management of Data, Jun. 1998, pp. 154-165.
W. A. Burkhard et al., "Some Approaches to Best-Match File Searching," Communications of the ACM, vol. 16, No. 4, Apr. 1973.
Eyal Kushilevitz et al., "Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces," Proceedings of the 30th annual ACM Symposium on the Theory of computing, 1998, pp. 457-474, vol. 30, No. 2.
J. Nievergelt et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," ACM Transactions on Database Systems, vol. 9, No. 1, pp. 38-71 (Mar. 1984).
Nevin Heintze, "Scalable Document Fingerprinting," Proc. USENIX Workshop on Electronic Commerce (1996).
Erling Wold et al., "Content-Based Classification, Search, and Retrieval of Audio," IEEE Multimedia, vol. 3, Issue 3, pp. 27-63 (1996).
Bir Bhanu et al., "Learning Feature Relevance and Similarity Metrics in Image Databases," Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 14-19 (1998).
A. Del Bimbo et al., "Using Weighted Spatial Relationships in Retrieval by Visual Contents," Image Description and Retrieval, pp. 161-192 (1998).
P. Indyk et al., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," Proceeding of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604-613 Jul. 21, 1999.
Marco La Cascia, "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web," Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 24-29 (1998).
Atsuo Yoshitaka et al., "A Survey on Content-Based Retrieval for Multimedia Databases," IEEE Transactions on Knowledge and Data Engineering, vol. 11, No. 1, pp. 81-93 (Jan./Feb. 1999).
Steve Lawrence et al., "Digital Libraries and Autonomous Citation Indexing," IEEE Computer, pp. 67-71 (Jun. 1999).
Akisato Kimura et al., "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," IEEE Conf on Acoustics, Speech and Signal Processing, (ICASSP '01), vol. 3, pp. 1429-1432, 2001.
Edgar Chavez et al., "Searching in Metric Spaces," ACM Computing Surveys, vol. 33, No. 3, pp. 273-321 (Sep. 2001).
Jaap Haitsma et al., "Robust Audio Hashing for Content Identification," Workshop on Content Based Multimedia Indexing, Brescia, Italy (Sep. 19-21, 2001).
Jaap Haitsma et al., "A Highly Robust Audio Fingerprinting System," Journal of New Music Research, 1744-5027, vol. 32, Issue 2, pp. 211-221 (2003).
Saul Schleimer et al., "Winnowing: Local Algorithms for Document Fingerprinting," ACM SIGMOD, Jun. 9-12, 2003.
Edward Chang et al., "Searching Near-Replicas of Images via Clustering," SPIE Symposium of Voice, Video and Data Communications, 1999.
Edward Y. Chang et al., "RIME: A Replicated Image Detector for the World-Wide Web," SPIE, 1998.
Hector Garcia-Molina et al., "Safeguarding and Charging for Information on the Internet," Proceedings of ICDE, 1998.
Sergey Brin et al., "Copy Detection Mechanisms for Digital Documents," Proceedings of ACM SIG-MOD, May 1995.
Stefan Berchtold "The x-tree: An Index Structure for High-Dimensional Data," Proceedings of the 22nd VLDB, Aug. 1996.
Norio Katayama et al., "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proceedings of ACM SIGMOD, May 1997.
John T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," Proceedings of ACM SIGMOD, Apr. 1981.
Myron Flickner et al., "Query by Image and Video Content: The QBIC System," IEEE Computer 28(9), pp. 23-32, 1995.
Amarnath Gupta et al., "Visual Information Retrieval," Communications of the ACM, vol. 40, No. 5, pp. 69-79, May 1997.
John R. Smith et al., "VisualSEEk: A fully automated content-based image query system," ACM Multimedia Conference, 1996.
David A. White et al., "Similarity Indexing: Algorithms and Performance," Proc. SPIE, vol. 2670, San Diego, 1996.
Norbert Beckmann et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," Proceedings of ACM Sigmod, May 1990.
A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proceedings of ACM Sigmod, Jun. 1984.
David A. White et al., "Similarity Indexing with the SS-tree'," Proceedings of the 12th ICDE, Feb. 1996.
King Lin et al., "The TV-Tree: An Index Structure for High-Dimensional Data," VLDB, Journal 3, No. 4, 1994, pp. 517-542.
Paolo Ciaccia et al., "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," Proceedings of the 23rd VLDB, Aug. 1997.
Nick Roussopoulos et al., Nearest Neighbor Queries, Proceedings of ACM Sigmod, May 1995.
C. Li et al., "An extensible hashing index for high-dimensional similarity search," Stanford Techncial Report, Aug. 1998.
Jon M. Kleinberg, "Two Algorithms for Nearest-Neighbor Search in High Dimensions," Proc 29th Stoc, Feb. 7, 1997.
U.S. Appl. No. 60/222,023, filed Jul. 31, 2000; Avery Li-Chun Wang and Julius O. Smith III, Inventors; Palo Alto, CA.
Liu et al., "An Investigation of Practical Approximate Nearest Neighbor Algorithms", Advances in Neural Information Processing Systems (NIPS) 2004.
K.L. Clarkson, "Nearest-Neighbor Searching and Metric Space Dimensions", Nearest-Neighbor Methods for Learning and Vision: Theory and Practice:, Apr. 2005.
Swaminathan et al., "Robust an Secure Image Hashing", IEEE Transactions on Information Forensics and Security, Jun. 2006, pp. 215-230, vol. 1, No. 2.
Burges et al., "Duplicate Detection and Audio Thumbnails with Audio Fingerprinting" [online]. 2004, [retrieved on Nov. 21, 2006]. Retrieved on the Internet: <URL: www.research.microsoft.com/cburges/tech_reports/tr-2004-19.pdf>, 5 pages.
Cana et al., "A Review of Algorithms for Audio Fingerprinting" [online]. 2002, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.iua.upf.es/mtg/publications/MMSP-2002-pcano.pdf>, 5 pages.
Haitsma and Kalker, "A Highly Robust Audio Fingerprinting System" [online]. 2002, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <url: www.ismir2002.ismir.net/proceedings/02-FP04-2.pdf>, 9 pages.
Jacobs et al., "Fast Multiresolution Image Querying" [online]. 1995, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.grail.cs.washington.edu/projects/query.pdf>, 10 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Ke et al., "Computer Vision for Music Identification" [online]. 2005, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.cs.cmu.edu/~yke/musicretrieval/cvpr2005-mr.pdf>, 8 pages.
Shazam, Shazam Enterprise Brings Music Recognition to Windows Mobile 5.0 Powered Smartphones [online]. 2006, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <URL: www.shazam.com/music/portal/sp/s/media-type/html/user/anon/page/default/template/pages/p/company_release30.html>, 1 page.
Stanford, "CS276 Information Retrieval and Web Mining" [online]. 2005, [retrieved on Nov. 16, 2006] Retrieved from the Internet: <URL: www.stanford.edu/class/cs276/handouts/lecture19.pdf>, 8 pages.
Stanford, "Data Mining: Associations" [online], 2002, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <URL: www.stanford.edu/class/cs206/cs206-2.pdf>, 11 pages.
Stollnitz et al., Wavelets for Computer Graphics: A Primer, Part 1: [online]. 1995, [retrieved on Nov. 21, 2008]. Retrieved from the Internet: <URL: www.grail.cs.washington.edu/pub/stoll/wavelet1.pdf>, 8 pages.
Stollnitz et al., Wavelets for Computer Graphics: A Primer, Part 2: [online]. 1995, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.grail.cs.washington.edu/pub/stoll/wavelet2.pdf>, 9 pages.
Yang, "MACS: Music Audio Characteristic Sequence Indexing for Similarity Retrieval", Oct. 21-24, 2001, New Peitz, New York.
Viola and Janes, Robust Real-Time Object Detection, Int. J. Computer Vision, 2002.
Burges et al., "Using Audio Fingerprinting for Duplicate Detection and Thumbnail Generation," Mar. 2005, 4 pages.
Lin, et al., Input Data Representation for Self-Organizing Map in Software Classification, Knowledge Acquisition and Modeling. 2009. KAM '09. Second International Symposium on vol. 2, Digital Object Identified: 10.11 09/KAM.2009.151 Publication Year: 2009. pp. 360-353.
Baiuja et al., "Content Fingerprinting Using Wavelets", 3rd European Conference on Visual Media Production, 2006, pp. 198-207.
Cohen et al., "Finding Interesting Associations without Support Pruning", IEEE Transactions on Knowledge and Data Engineering, 2001, pp. 64-78, vol. 13, issue 1.
Yang, Efficient Video Identification based on locality sensitive hashing and triangle inequality, National University of Singapore, 2005, pp. 1-64.
U.S. Appl. No. 60/304,647, filed Jul. 10, 2001.
U.S. Appl. No. 60/281,881, filed Apr. 5, 2001.
Metadata Mediation: Representation and Protocol, Tsuyoshi Sakata, Hiroyuki Tada, Tomohisa Ohtake, Digital Vision Laboratories, 7-3-37 Akasaka, Minato, Tokyo, Japan.
U.S. Appl. No. 13/800,573, filed Mar. 13, 2013.
U.S. Appl. No. 13/800,890, filed Mar. 13, 2013.
U.S. Appl. No. 13/829,717, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,447, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,626, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,986, filed Mar. 14, 2013.
U.S. Appl. No. 13/842,068, filed Mar. 15, 2013.
U.S. Appl. No. 13/338,079, filed Dec. 27, 2011.
U.S. Appl. No. 60/134,782, filed May 19, 1999.
U.S. Appl. No. 60/193,948, filed Mar. 31, 2000.
U.S. Appl. No. 60/195,535, filed Apr. 7, 2000.
U.S. Appl. No. 60/206,384, filed May 23, 2000.
U.S. Appl. No. 60/221,843, filed Jul. 28, 2000.
U.S. Appl. No. 60/133,247, filed May 5, 1999.
U.S. Appl. No. 60/155,064, filed Sep. 21, 1999.
U.S. Appl. No. 60/218,824, filed Jul. 18, 2000.
Indyk, Piotr et al., "Locality-Preserving Hashing in Multidimensional Spaces," Feb. 25, 1997.
Gibson, David, "Name That Clip: Music retrieval using audio clips," Aug. 19, 1999.
Declaration of David A. Gibson, Inter Partes Review of U.S. Patent 7,174,293, Aug. 30, 2013.
Declaration of David A. Gibson, Inter Partes Review of U.S. Patent 7,783,489, Aug. 30, 2013.
Intersil, "Glossary of Communication Terms," Dec. 1996.
Declaration of Dr. Ton Kalker, Inter Partes Review of U.S. Patent 7,174,293, Aug. 30, 2013.
Declaration of Dr. Ton Kalker, Inter Partes Review of U.S. Patent 7,783,489, Aug. 30, 2013.
Ardizzone, Edoardo et al., "Motion and Color-Based Video Indexing and Retrieval," Universita di palermo, Departimento di Ingegneria Elettrica, pp. 135-139, Viale delle Scienze, Palermo, Italy, IEEE 1996.
Deng, Yining et al., "Content-based Search of Video Using Color, Texture, and Motion," Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA pp. 534-537, IEEE 1997.
Fang, Min et al., "Computing Iceberg Queries Efficiently," Dept. of Computer Science, Stanford, CA, Paper No. 234, pp1-25.
Flickner, Myron et al., "Query by Image and Video Content: The QBIC System," IBM Almaden Research Center, Sep. 1995, pp. 23-32, IEEE 1995.
Gargi, U et al., "Performance Characterization and Comparison of Video Indexing Algorithms," Dept. of Computer Science & Engineering, The Pennsylvania State University, University Park, PA.
Gionis, Aristides et al., "Similarity Search in High Dimensions via Hashing," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 518-529, Proceeding of the 25th VLDB Conference, Edinburgh, Scotland, 1999.
Indyk, Piotr et al., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality" (preliminary version) Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13 & i-vii, Jul. 21, 1999.
Iyengar, Giridharan et al., "Models for automatic classification of video sequences," MIT Media Laboratory, Cambridge, MA.
Jain, Anil K., et al., "Image Retrieval using Color and Shape," Dept. of Computer Science, Michigan State University, Eas Lansing, MI, pp. 1-24, May 15, 1995.
Ogle, Virginia E., et al., "Chabot: Retrieval from a Relational Database of Images," University of California at Berkeley, Computer pp. 40-48, IEEE 1995.
Pentland, A. et al., "Photobook: Content-Based Manipulation of Image Databases," Perceptual Computing Section, the Media Laboratory, Massachusetts Institute of Tech., International Journal of Computer Vision 18(3), pp. 233-254 (1996), 1996 Kluwer Academic Publishers. Manuf. In the Netherlands.
Shivakumar, Narayanan et al., "Scam: a Copy Detection Mechanism for Digital Documents," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13.
Shivakumar, Narayanan et al., "Building a Scalable and Accurate Copy Detection Mechanism," Dept. of Computer Science, Stanford University, Stanford, CA.
Srihari, Rohini K., "Automatic Indexing and Content-Based Retrieval of Captioned Images," State University of New York, Buffalo, Theme Feature, pp. 49-56, Sept. 1995, IEEE 1995.
Swain, Michael and Ballard, Dana H., "Color Indexing," International Journal of Computer Vision 7:1, p. 11-32 (1991), 1991 Kluwer Academic Publishers. Manuf. in the Netherlands.
Wactlar, Howard D. et al., "Intelligent Access to Digital Video: Informedia Project," Carnegie Mellon University, Digital Library Initiative: Carnegie Mellon University, Computer, pp. 46-52, IEEE 1996.
Yeo, Boon-Lock et al., "Rapid Scene Analysis on Compressed Video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, No. 6, pp. 533-544, Dec. 1995, Dept. of Electrical Engineering, Princeton University, Princeton, NJ, IEEE Log No. 9415901.
Indyk, Piotr et al., "Finding pirated video sequences on the Internet," Dept. of Computer Science, Stanford University, Palo Alto, CA, Paper No. 199.

* cited by examiner



WORK @t1
WORK @t2
FEATURE EXTRACTION OPERATION(S)
122
FEATURE TO WORK ID TAGGING OPERATION(S)
124
DATABASE GENERATION OPERATION(S)
120
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140
WID INFORMATION
110
150
FEATURE (VECTOR) LOOKUP OPERATION(S)
114
116
FEATURE(S) (VECTOR)
WORK ID
112
138
160
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WID-ACTION INFORMATION
130
DATABASE GENERATION OPERATION(S)
134
136
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
132
ACTION INITIATION OPERATION(S)
170
100

FIGURE 1

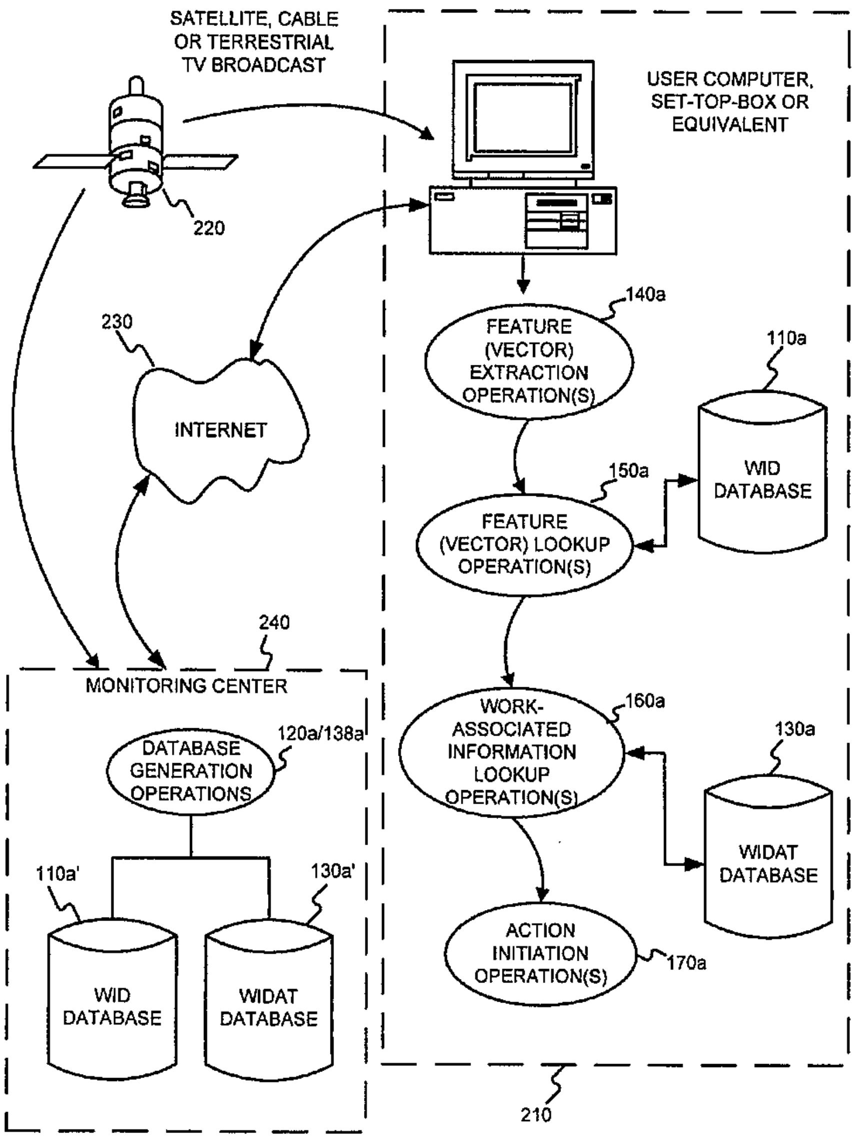
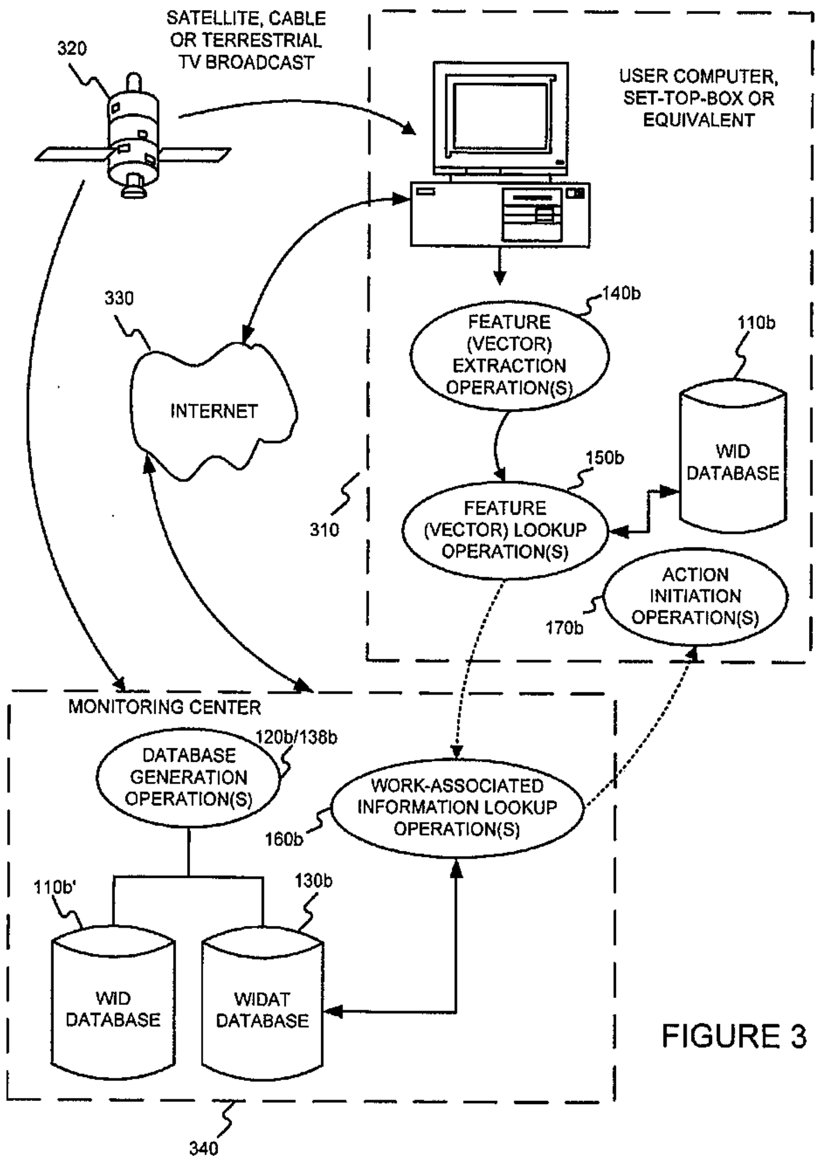

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
220
140a
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
230
110a
INTERNET
150a
WID
DATABASE
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
240
MONITORING CENTER
160a
WORK-
ASSOCIATED
INFORMATION
LOOKUP
OPERATION(S)
DATABASE
GENERATION
OPERATIONS
120a/138a
130a
WIDAT
DATABASE
110a'
130a'
ACTION
INITIATION
OPERATION(S)
170a
WID
DATABASE
WIDAT
DATABASE
210

FIGURE 2



SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
320
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
330
INTERNET
140b
FEATURE (VECTOR) EXTRACTION OPERATION(S)
110b
WID DATABASE
150b
FEATURE (VECTOR) LOOKUP OPERATION(S)
310
ACTION INITIATION OPERATION(S)
170b
MONITORING CENTER
120b/138b
DATABASE GENERATION OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
160b
110b'
130b
WID DATABASE
WIDAT DATABASE
340

FIGURE 3

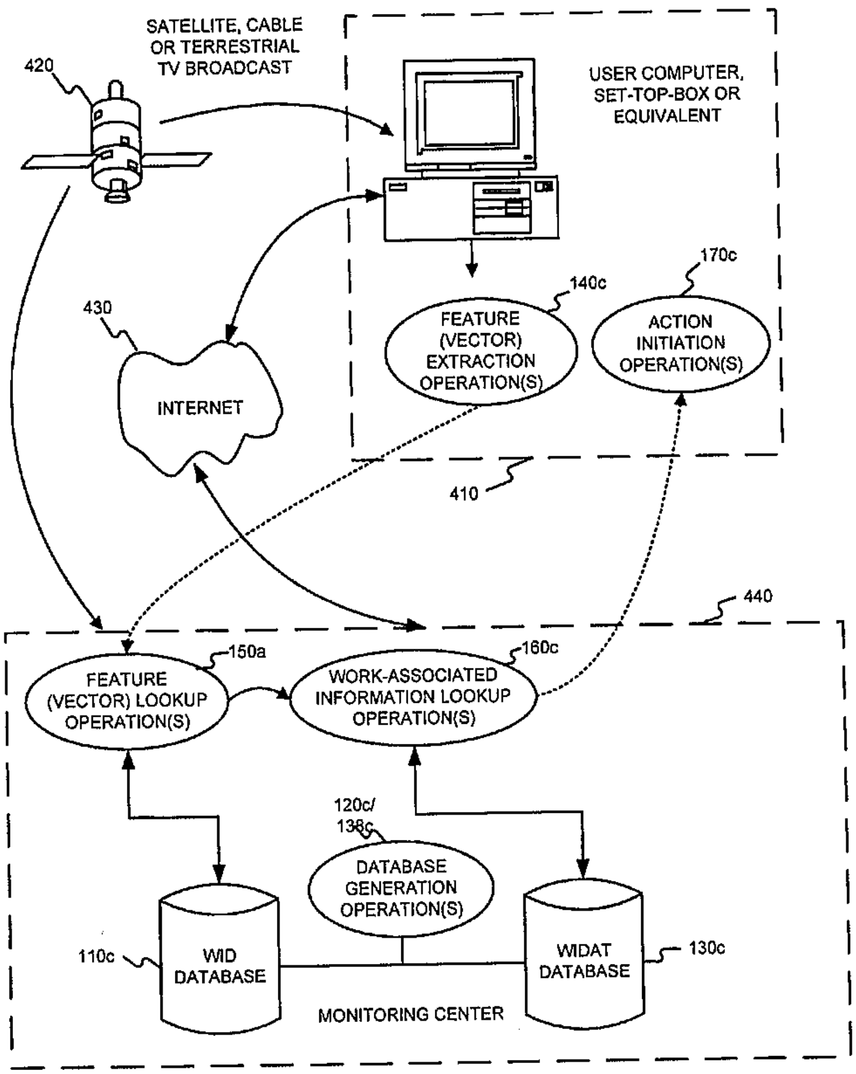

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
420
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
170c
140c
430
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
ACTION
INITIATION
OPERATION(S)
INTERNET
410
440
150a
160c
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
120c/
130c
DATABASE
GENERATION
OPERATION(S)
110c
WID
DATABASE
WIDAT
DATABASE
130c
MONITORING CENTER

FIGURE 4

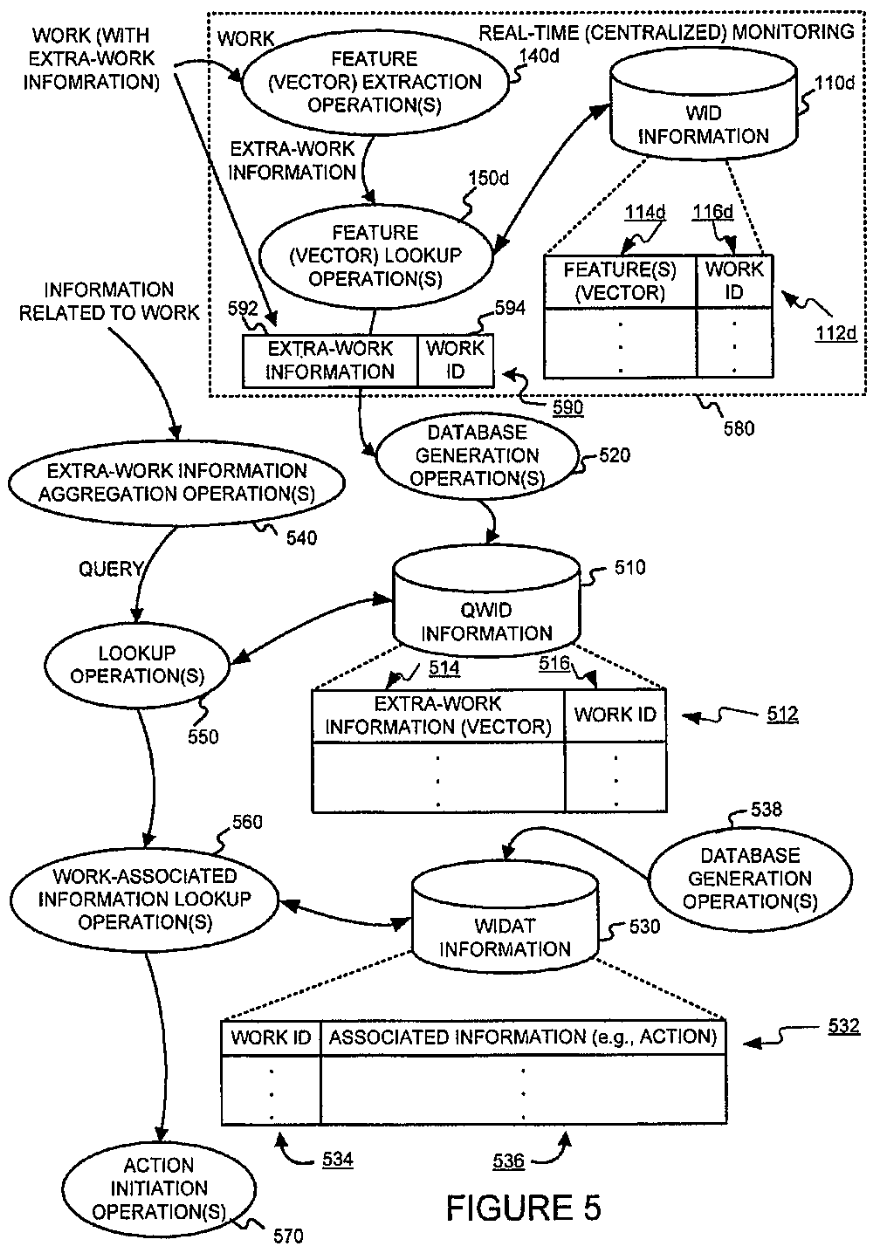

WORK (WITH EXTRA-WORK INFOMRATION)
WORK
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140d
REAL-TIME (CENTRALIZED) MONITORING
WID INFORMATION
110d
EXTRA-WORK INFORMATION
150d
FEATURE (VECTOR) LOOKUP OPERATION(S)
114d
116d
FEATURE(S) (VECTOR)
WORK ID
112d
INFORMATION RELATED TO WORK
592
594
EXTRA-WORK INFORMATION
WORK ID
590
580
DATABASE GENERATION OPERATION(S)
520
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
540
QUERY
510
QWID INFORMATION
LOOKUP OPERATION(S)
550
514
516
EXTRA-WORK INFORMATION (VECTOR)
WORK ID
512
538
560
DATABASE GENERATION OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WIDAT INFORMATION
530
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
532
534
536
ACTION INITIATION OPERATION(S)
570

FIGURE 5

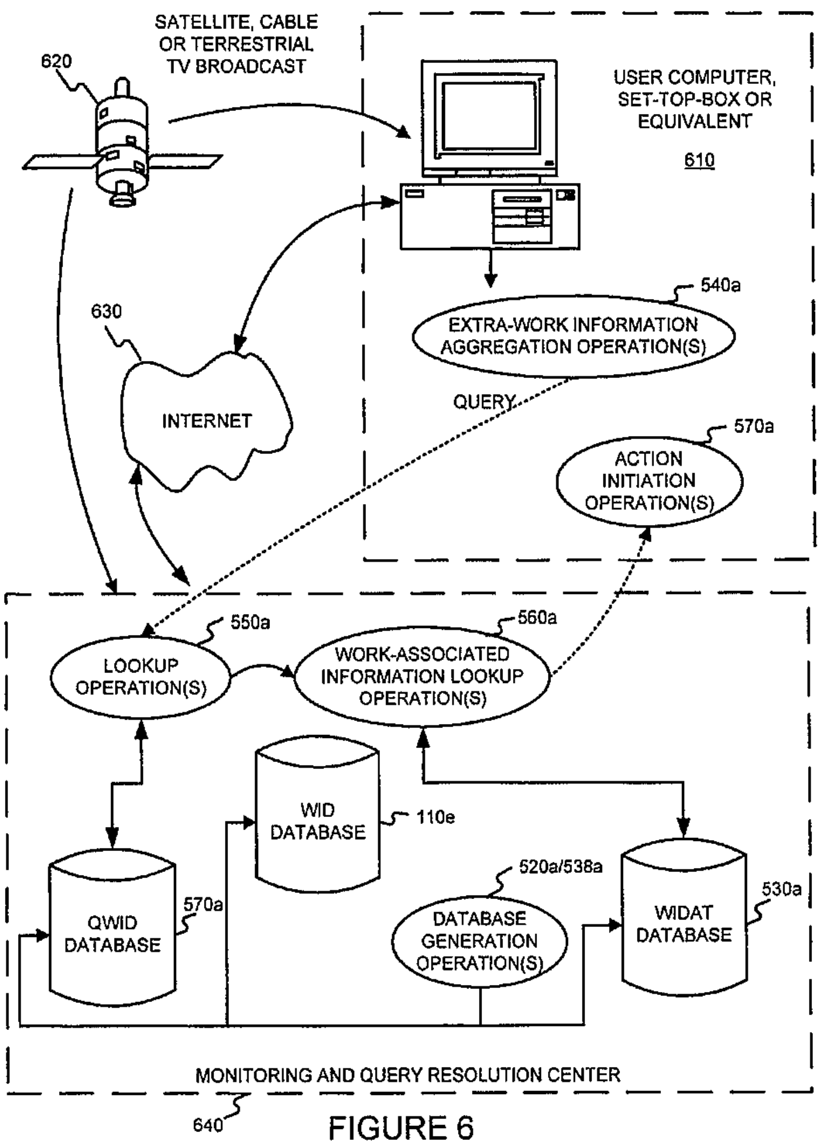
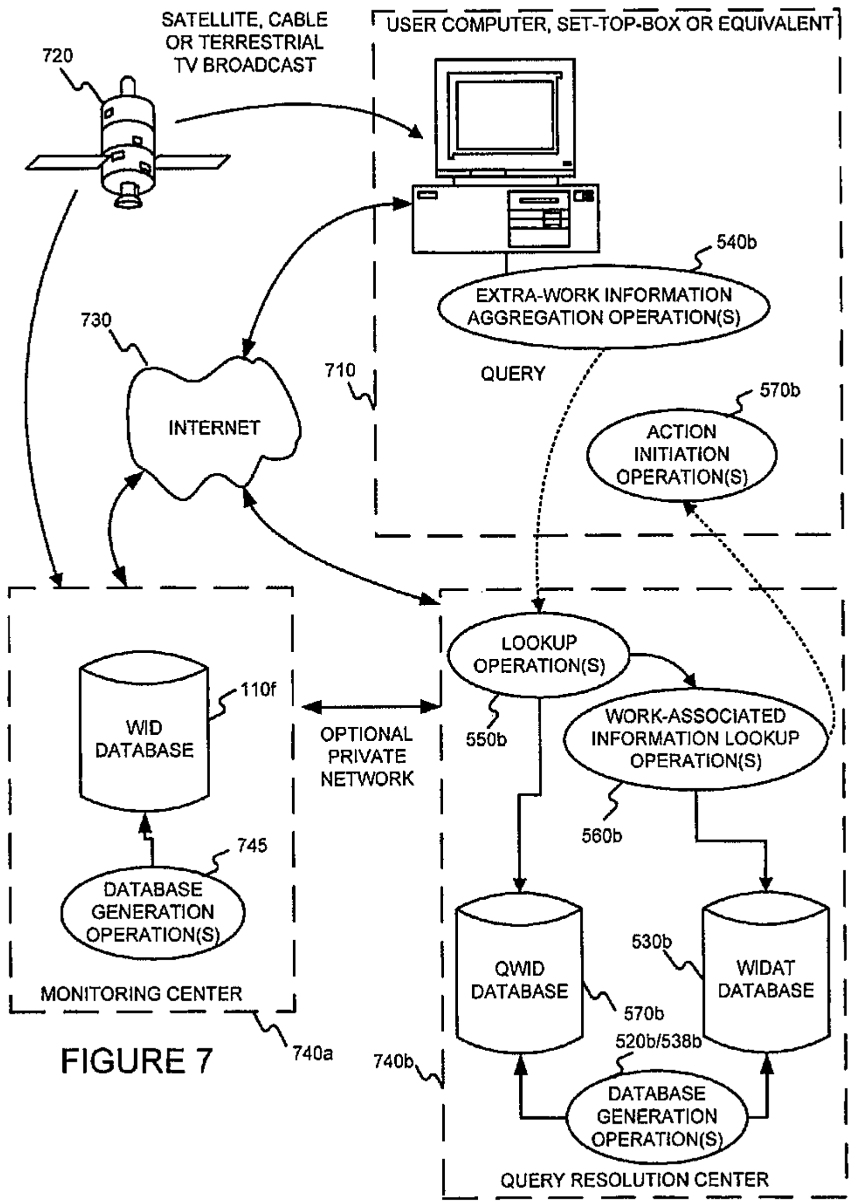

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
620
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
610
540a
EXTRA-WORK INFORMATION
AGGREGATION OPERATION(S)
630
INTERNET
QUERY
570a
ACTION
INITIATION
OPERATION(S)
550a
560a
LOOKUP
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
WID
DATABASE
110e
520a/538a
QWID
DATABASE
570a
DATABASE
GENERATION
OPERATION(S)
WIDAT
DATABASE
530a
MONITORING AND QUERY RESOLUTION CENTER
640

FIGURE 6



SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
720
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
540b
EXTRA-WORK INFORMATION
AGGREGATION OPERATION(S)
730
INTERNET
710
QUERY
570b
ACTION
INITIATION
OPERATION(S)
LOOKUP
OPERATION(S)
550b
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
560b
110f
WID
DATABASE
OPTIONAL
PRIVATE
NETWORK
745
DATABASE
GENERATION
OPERATION(S)
MONITORING CENTER
740a
QWID
DATABASE
570b
530b
WIDAT
DATABASE
520b/538b
DATABASE
GENERATION
OPERATION(S)
740b
QUERY RESOLUTION CENTER

FIGURE 7

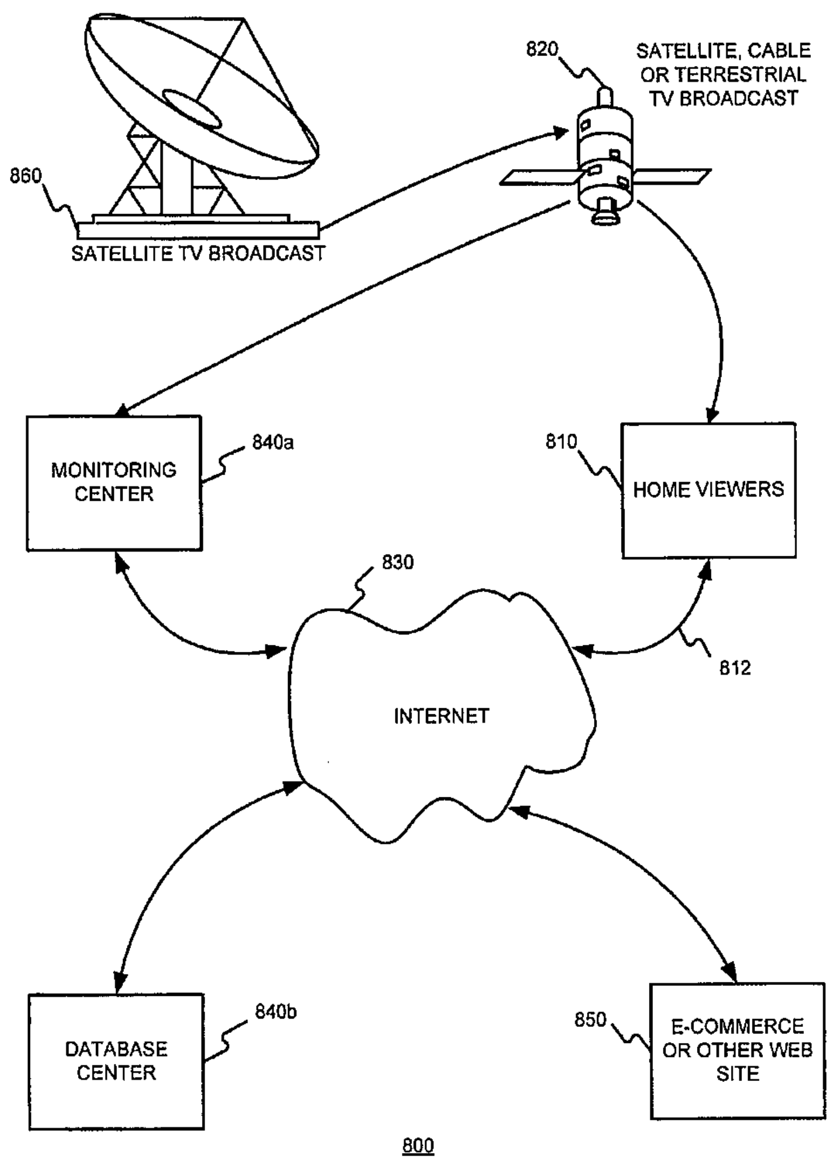

860
SATELLITE TV BROADCAST
820
SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
MONITORING CENTER
840a
810
HOME VIEWERS
830
INTERNET
812
DATABASE CENTER
840b
850
E-COMMERCE OR OTHER WEB SITE
800

FIGURE 8

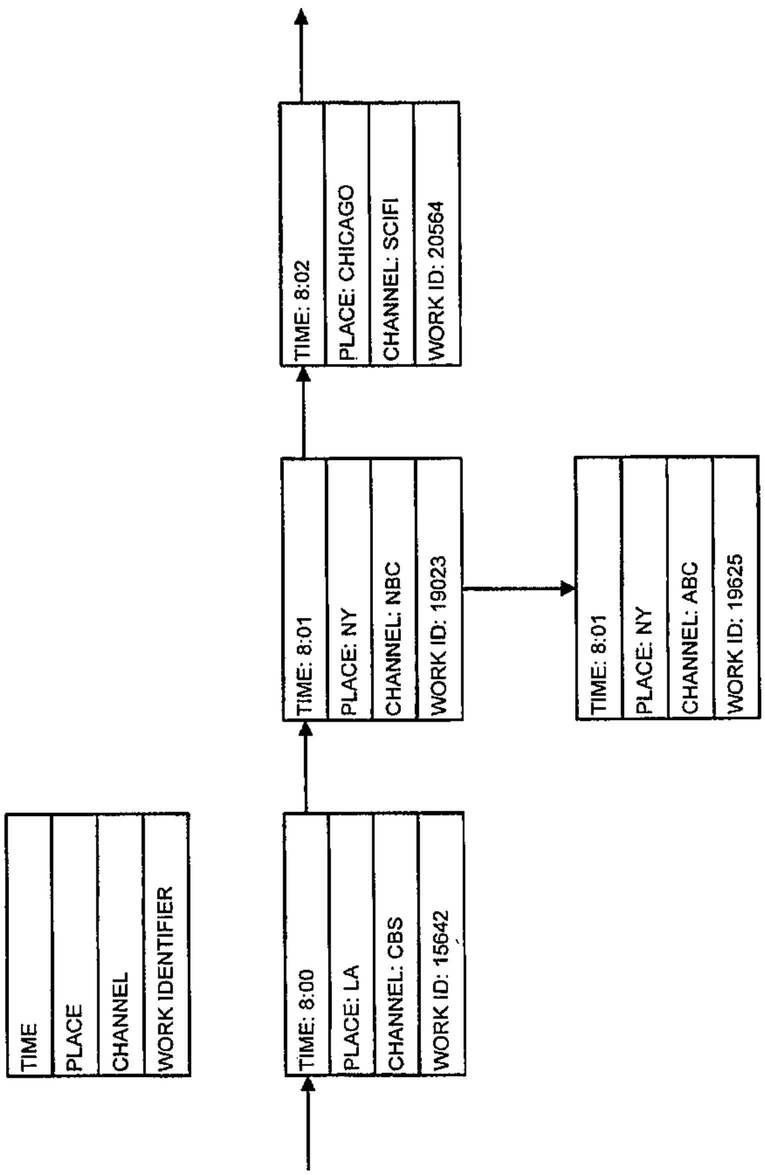

TIME
PLACE
CHANNEL
WORK IDENTIFIER
TIME: 8:00
PLACE: LA
CHANNEL: CBS
WORK ID: 15642
TIME: 8:01
PLACE: NY
CHANNEL: NBC
WORK ID: 19023
TIME: 8:01
PLACE: NY
CHANNEL: ABC
WORK ID: 19625
TIME: 8:02
PLACE: CHICAGO
CHANNEL: SCIFI
WORK ID: 20564

FIGURE 9

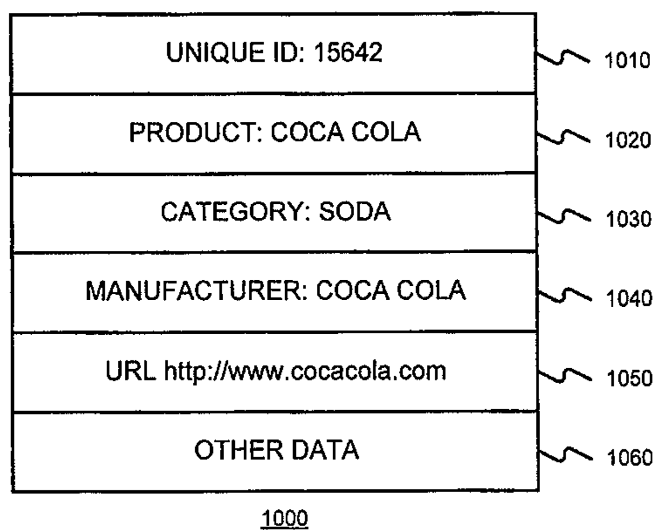

UNIQUE ID: 15642
1010
PRODUCT: COCA COLA
1020
CATEGORY: SODA
1030
MANUFACTURER: COCA COLA
1040
URL http://www.cocacola.com
1050
OTHER DATA
1060
1000

FIGURE 10

METHOD FOR USING EXTRACTED FEATURES FROM AN ELECTRONIC WORK

§0. RELATED APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 11/977,202 (incorporated herein by reference), titled "IDENTIFYING WORKS, USING A SUB-LINEAR TIME SEARCH, SUCH AS AN APPROXIMATE NEAREST NEIGHBOR SEARCH, FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET", filed Oct. 23, 2007, and listing Ingemar J. Cox as the inventor, which is a continuation of U.S. patent application Ser. No. 11/445,928 (incorporated herein by reference), titled "USING FEATURES EXTRACTED FROM AN AUDIO AND/OR VIDEO WORK TO OBTAIN INFORMATION ABOUT THE WORK," filed on Jun. 2, 2006, and listing Ingemar J. Cox as the inventor, which is a continuation-in-part of U.S. patent application Ser. No. 09/950,972 (incorporated herein by reference, issued as U.S. Pat. No. 7,058,223 on Jun. 6, 2006), titled "IDENTIFYING WORKS FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET," filed on Sep. 13, 2001, and listing Ingemar J. Cox as the inventor, which application claims benefit to the filing date of provisional patent application Ser. No. 60/232,618 (incorporated herein by reference), titled "Identifying and linking television, audio, print and other media to the Internet", filed on Sep. 14, 2000 and listing Ingemar J. Cox as the inventor.

§1. BACKGROUND OF THE INVENTION

§1.1 Field of the Invention

The present invention concerns linking traditional media to new interactive media, such as that provided over the Internet for example. In particular, the present invention concerns identifying a work (e.g., content or an advertisement delivered via print media, or via a radio or television broadcast) without the need to modify the work.

§1.2 Related Art

§1.2.1 Opportunities Arising from Linking Works Delivered Via Some Traditional Media Channel or Conduit to a More Interactive System

The rapid adoption of the Internet and associated World Wide Web has recently spurred interest in linking works, delivered via traditional media channels or conduits, to a more interactive system, such as the Internet for example. Basically, such linking can be used to (a) promote commerce, such as e-commerce, and/or (b) enhance interest in the work itself by facilitating audience interaction or participation. Commerce opportunities include, for example, facilitating the placement of direct orders for products, providing product coupons, providing further information related to a product, product placement, etc.

In the context of e-commerce, viewers could request discount vouchers or coupons for viewed products that are redeemable at the point of purchase. E-commerce applications also extend beyond advertisements. It is now common for television shows to include product placements. For example, an actor might drink a Coke rather than a Pepsi brand of soda, actors and actresses might wear designer-labeled clothing such as Calvin Klein, etc. Viewers may wish to purchase similar clothing but may not necessarily be able to

identify the designer or the particular style directly from the show. However, with an interactive capability, viewers would be able to discover this and other information by going to an associated Web site. The link to this Web site can be automatically enabled using the invention described herein.

In the context of facilitating audience interaction or participation, there is much interest in the convergence of television and computers. Convergence encompasses a very wide range of capabilities. Although a significant effort is being directed to video-on-demand applications, in which there is a unique video stream for each user of the service, as well as to transmitting video signals over the Internet, there is also interest in enhancing the television viewing experience. To this end, there have been a number of experiments with interactive television in which viewers can participate in a live broadcast. There are a variety of ways in which viewers can participate. For example, during game shows, users can answer the questions and their scores can be tabulated. In recent reality-based programming such as the ABC television game show, "Big Brother", viewers can vote on contestants who must leave the show, and be eliminated from the competition.

§1.2.2 Embedding Work Identifying Code or Signals within Works

Known techniques of linking works delivered via traditional media channels to a more interactive system typically require some type of code, used to identify the work, to be inserted into the work before it is delivered via such traditional media channels. Some examples of such inserted code include (i) signals inserted into the vertical blanking interval ("VBI") lines of a (e.g., NTSC) television signal, (ii) watermarks embedded into images, (iii) bar codes imposed on images, and (iv) tones embedded into music.

The common technical theme of these proposed implementations is the insertion of visible or invisible signals into the media that can be decoded by a computer. These signals can contain a variety of information. In its most direct form, the signal may directly encode the URL of the associated Web site. However, since the alphanumeric string has variable length and is not a particularly efficient coding, it is more common to encode a unique ID. The computer then accesses a database, which is usually proprietary, and matches the ID with the associated web address. This database can be considered a form of domain name server, similar to those already deployed for network addresses. However, in this case, the domain name server is proprietary and the addresses are unique ID's.

There are two principal advantages to encoding a proprietary identifier into content. First, as previously mentioned, it is a more efficient use of the available bandwidth and second, by directing all traffic to a single Web site that contains the database, a company can maintain control over the technology and gather useful statistics that may then be sold to advertisers and publishers.

As an example of inserting signals into the vertical blanking interval lines of a television signal, RespondTV of San Francisco, Calif. embeds identification information into the vertical blanking interval of the television signal. The VBI is part of the analog video broadcast that is not visible to television viewers. For digital television, it may be possible to encode the information in, for example, the motion picture experts group ("MPEG") header. In the USA, the vertical blanking interval is currently used to transmit close-captioning information as well as other information, while in the UK, the VBI is used to transmit teletext information. Although the close captioning information is guaranteed to be transmitted into the home in America, unfortunately, other information is not. This is because ownership of the vertical blanking interval is disputed by content owners, broadcasters and local television operators.

As an example of embedding watermarks into images, Digimarc of Tualatin, Oreg. embeds watermarks in print media. Invisible watermarks are newer than VBI insertion, and have the advantage of being independent of the method of broadcast. Thus, once the information is embedded, it should remain readable whether the video is transmitted in NTSC, PAL or SECAM analog formats or newer digital formats. It should be more reliable than using the vertical blanking interval in television applications. Unfortunately, however, watermarks still require modification of the broadcast signal which is problematic for a number of economic, logistical, legal (permission to alter the content is needed) and quality control (the content may be degraded by the addition of a watermark) reasons.

As an example of imposing bar codes on images, print advertisers are currently testing a technology that allows an advertisement to be shown to a camera, scanner or bar code reader that is connected to a personal computer ("PC"). The captured image is then analyzed to determine an associated Web site that the PC's browser then accesses. For example, GoCode of Draper, Utah embeds small two-dimensional bar codes for print advertisements. The latter signal is read by inexpensive barcode readers that can be connected to a PC. AirClic of Blue Bell, Pa. provides a combination of barcode and wireless communication to enable wireless shopping through print media. A so-called "CueCat" reads bar codes printed in conjunction with advertisements and articles in Forbes magazine. Similar capabilities are being tested for television and audio media.

Machine-readable bar codes are one example of a visible signal. The advantage of this technology is that it is very mature. However, the fact that the signal is visible is often considered a disadvantage since it may detract from the aesthetic of the work delivered via a traditional media channel or conduit.

As an example of embedding tones into music, Digital Convergence of Dallas, Tex. proposes to embed identification codes into audible music tones broadcast with television signals.

All the foregoing techniques of inserting code into a work can be categorized as active techniques in that they must alter the existing signal, whether it is music, print, television or other media, such that an identification code is also present. There are several disadvantages that active systems share. First, there are aesthetic or fidelity issues associated with bar codes, audible tones and watermarks. More importantly, all media must be processed, before it is delivered to the end user, to contain these active signals. Even if a system is enthusiastically adopted, the logistics involved with inserting bar codes or watermarks into, say every printed advertisement, are formidable.

Further, even if the rate of adoption is very rapid, it nevertheless remains true that during the early deployment of the system, most works will not be tagged. Thus, consumers that are early-adopters will find that most media is not identified. At best, this is frustrating. At worst, the naive user may conclude that the system is not reliable or does not work at all. This erroneous conclusion might have a very adverse effect on the adoption rate.

Further, not only must there be modification to the production process, but modifications must also be made to the equipment in a user's home. Again, using the example of watermarking of print media, a PC must be fitted with a camera and watermark detection software must be installed. In the case of television, the detection of the identification signal is likely to occur at the set-top-box—this is the equipment provided by the local cable television or satellite broadcasting company. In many cases, this may require modifications to the hardware, which is likely to be prohibitively expensive. For example, the audible tone used by Digital Convergence to recognize television content, must be fed directly into a sound card in a PC. This requires a physical connection between the television and the PC, which may be expensive or at least inconvenient, and a sound card may have to be purchased.

§1.2.3 Unmet Needs

In view of the foregoing disadvantages of inserting an identification code into a work, thereby altering the existing signal, there is a need for techniques of identifying a work without the need of inserting an identification code into a work. Such an identification code can then be used to invoke a work-related action, such as work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§2. SUMMARY OF THE INVENTION

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable programs for linking a media work to an action. Such embodiments might (a) extract features from the media work, (b) determine an identification of the media work based on the features extracted using a sub-linear time search, such as an approximate nearest neighbor search for example, and (c) determine an action based on the identification of the media work determined. In some embodiments consistent with the present invention, the media work is an audio signal. The audio signal might be obtained from a broadcast, or an audio file format. In other embodiments consistent with the present invention, the media work is a video signal. The video signal might be obtained from a broadcast, or a video file format.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§3. BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work.

FIG. 2 is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 3 is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 4 is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 5 is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work.

FIG. 6 is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 8 is a block diagram illustrating an environment in which the present invention may operate.

FIG. 9 is an exemplary data structure in which extra-work information is associated with a work identifier.

FIG. 10 is an exemplary data structure including work-related actions.

§4. DETAILED DESCRIPTION

The present invention may involve novel methods, apparatus and data structures for identifying works without the need of embedding signals therein. Once identified, such information can be used to determine a work-related action. The following description is presented to enable one skilled in the art to make and use the invention, and is provided in the context of particular embodiments and methods. Various modifications to the disclosed embodiments and methods will be apparent to those skilled in the art, and the general principles set forth below may be applied to other embodiments, methods and applications. Thus, the present invention is not intended to be limited to the embodiments and methods shown and the inventors regard their invention as the following disclosed methods, apparatus, data structures and any other patentable subject matter to the extent that they are patentable.

§4.1 FUNCTIONS

The present invention functions to identify a work without the need of inserting an identification code into a work. The present invention may do so by (i) extracting features from the work to define a feature vector, and (ii) comparing the feature vector to feature vectors associated with identified works. Alternatively, or in addition, the present invention may do so by (i) accepting extra-work information, such as the time of a query or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and (ii) use such extra-work information to lookup an identification of the work. In either case, an identification code may be used to identify the work.

The present invention may then function to use such an identification code to initiate a work-related action, such as for work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§4.2 EMBODIMENTS

As just introduced in §4.1 above, the present invention may use intra-work information and/or extra-work information to identify a work. Once identified, such identification can be used to initiate an action, such as an action related to commerce, or facilitating audience participation or interaction. Exemplary embodiments of the present invention, in which work is recognized or identified based on intra-work information, are described in §4.2.1. Then, exemplary embodiments of the present invention, in which work is recognized or identified based on extra-work information, are described in §4.2.2.

§4.2.1 Embodiments in which Work is Recognized Based on Intra-Work Information, Such as a Feature Vector

Operations related to this embodiment are described in §4.2.1.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.1.2.

§4.2.1.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work. As shown, a work-identification information storage **110** may include a number of items or records **112**. Each item or record **112** may associate a feature vector of a work **114** with a, preferably unique, work identifier **116**. The work-identification information storage **110** may be generated by a database generation operation(s) **120** which may, in turn, use a feature extraction operation(s) **122** to extract features from a work at a first time (WORK.sub.@t1), as well as a feature-to-work identification tagging operation(s) **124**.

Further, work identifier-action information storage **130** may include a number of items or records **132**. Each item or record **132** may associate a, preferably unique, work identifier **134** with associated information **136**, such as an action for example. The work identifier-action information storage **130** may be generated by a database generation operation(s) **138** which may, for example, accept manual entries.

As can be appreciated from the foregoing, the work-information storage **110** records **112** and the work identification-action **130** records **132** can be combined into a single record. That is, there need not be two databases. A single database is also possible in which the work identifier, or a feature vector extracted from the work, serves as a key and the associated field contains work-related information, such as a URL for example.

The feature extraction operation(s) **140** can accept a work, such as that being rendered by a user, at a second time (WORK.sub.@t2), and extract features from that work. The extracted features may be used to define a so-called feature vector.

The extracted features, e.g., as a feature vector, can be used by a feature (vector) lookup operation(s) **150** to search for a matching feature vector **114**. If a match, or a match within a predetermined threshold is determined, then the associated work identifier **116** is read.

The read work identifier can then be used by a work-associated information lookup operation(s) **160** to retrieve associated information, such as an action, **136** associated with the work identifier. Such information **136** can then be passed to action initiation operation(s) **170** which can perform some action based on the associated information **136**.

§4.2.1.1.1 Exemplary Techniques for Feature Extraction

When the user initiates a request, the specific television or radio broadcast or printed commercial, each of which is referred to as a work, is first passed to the feature extraction operation. The work may be an image, an audio file or some portion of an audio signal or may be one or more frames or fields of a video signal, or a multimedia signal. The purpose of the feature extraction operation is to derive a compact representation of the work that can subsequently be used for the purpose of recognition. In the case of images and video, this feature vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n.times.n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, wavelet and or discrete cosine transforms. It might involve principal component analysis. It might also be a combination of these. For television and audio signals, recognition might also rely on a temporal sequence of feature vectors. The recognition literature contains many different representations. For block-based methods, blocks may be accessed at pseudo-random locations in each frame or might have a specific structure. For audio, common feature vectors are based on Fourier frequency decompositions, but other representations are possible. See, e.g., R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, 1973). See also K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Ed. (Academic Press, New York, 1990). (These references are incorporated herein by reference.)

As previously stated, one object of the vector extraction stage is to obtain a more concise representation of the frame. For example, each video frame is initially composed of 480.times.720 pixels which is equivalent to 345,600 pixels or 691,200 bytes. In comparison, an exemplary feature vector might only consist of 1 Kbyte of data.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e., horizontal and or vertical translation, or may undergo lossy compression such as by MPEG-2. It is advantageous that these and other processes do not adversely affect the extracted vectors. For still images there has been considerable work on determining image properties that are invariant to affine and other geometric distortions. For example, the use of Radon and Fourier-Mellin transforms have been proposed for robustness against rotation, scale and translation, since these transforms are either invariant or bare a simple relation to the geometric distortions. See, e.g., C. Lin, M. Wu, Y. M. Lui, J. A. Bloom, M. L. Miller, I. J. Cox, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," IEEE Transactions on Image Processing (2001). See also, U.S. Pat. Nos. 5,436,653, 5,504,518, 5,582,246, 5,612,729, and 5,621,454. (Each of these references is incorporated herein by reference.)

§4.2.1.1.2 Exemplary Techniques for Database Generation and Maintenance

A number of possibilities exist for generating and maintaining work identification (WID) and identification-action translation (WIDAT) databases. However, in all cases, works of interest are processed to extract a representative feature vector and this feature vector is assigned a unique identifier. This unique identifier is then entered into the work identification (WID) database **110** as well as into the WIDAT database **130** together with all the necessary associated data. This process is referred to as tagging. For example, in the case of an advertisement, the WIDAT database **130** might include the manufacturer (Ford), the product name (Taurus), a product category (automotive) and the URL associated with the Ford Taurus car together with the instruction to translate the query into the associated URL.

The determination of all works of interest and subsequent feature vector extraction and tagging depends on whether content owners are actively collaborating with the entity responsible for creating and maintaining the database. If there is no collaboration, then the database entity must collect all works of interest and process and tag them. While this is a significant effort, it is not overwhelming and is certainly commercially feasible. For example, competitive market research firms routinely tabulate all advertisements appearing in a very wide variety of print media. Newspapers and magazines can be scanned in and software algorithms can be applied to the images to identify likely advertisements. These possible advertisements can then be compared with advertisements already in the WID database **110**. If there is a match, nothing further need be done. If there is not a match, the image can be sent to a human to determine if the page does indeed contain an advertisement. If so, the operator can instruct the computer to extract the representative feature vector and assign it a unique identifier. Then, the operator can insert this information into the content identification database and as well as update the corresponding WIDAT database **130** with all the necessary associated data. This is continually performed as new magazines and papers include new advertisements to maintain the databases. This is a cost to the database entity. Television and radio broadcasts can also be monitored and, in fact, broadcast monitoring is currently performed by companies such as Nielsen Media research and Competitive Media Reporting. Television and radio broadcasts differ from print media in the real-time nature of the signals and the consequent desire for real-time recognition.

In many cases, advertisers, publishers and broadcasters may wish to collaborate with the database provider. In this case, feature extraction and annotation and/or extra-work information may be performed by the advertiser, advertisement agency, network and/or broadcaster and this information sent to the database provider to update the database. Clearly, this arrangement is preferable from the database provider's perspective. However, it is not essential.

§4.2.1.1.3. Exemplary Techniques for Matching Extracted Features with Database Entries

The extracted feature vector is then passed to a recognition (e.g., feature look-up) operation, during which, the vector is compared to entries of known vectors **114** in a content identification (WID) database **110**. It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search might not be possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used including mutual information, Euclidean distance and Lp-norms. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and false negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match. See, e.g., R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, 1973). See also, U.S. Pat. No. 3,919,474 by W. D. Moon, R. J. Weiner, R. A. Hansen and R. N. Linde, entitled

"Broadcast Signal Identification System". (Each of these references is incorporated herein by reference.)

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. Unfortunately, binary search is not possible when taking a noisy signal and trying to find the most similar reference signal. This problem is one of nearest neighbor search in a (high-dimensional) feature space. In previous work, it was not uncommon to perform a linear search of all N entries, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this search can be computationally very expensive.

Other forms of matching include those based on clustering, kd-trees, vantage point trees and excluded middle vantage point forests are possible and will be discussed in more detail later. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, (Jan. 15, 1999). See also, P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point.

If the extracted vector "matches" a known vector in the content identification database, then the work has been identified. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when the vector extracted from a work is not matched to the database even though the work is present in the database. There are several reasons why a works feature vector may fail to match a feature vector database entry. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will often contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible.

Finally, there is the case where the observed work is not present in the database. In this case, the work can be sent to an operator for identification and insertion in the database.

§4.2.1.1.4 Exemplary Work Based Actions

Assuming that the work is correctly identified, then the identifier can be used to retrieve associated information from the second work identification-action translation (WIDAT) database **130** that contains information **136** associated with the particular work **134**. This information may simply be a corresponding URL address, in which case, the action can be considered to be a form of network address translation. However, in general, any information about the work could be stored therein, together with possible actions to be taken such as initiating an e-commerce transaction. After looking up the work identifier **134** in the WIDAT database **130**, an action is performed on behalf of the user, examples of which has been previously described.

In addition to using the system to allow audience members of a work to connect to associated sites on the Internet, a number of other uses are possible. First, the work identification database **130** allows competitive market research data to be collected (e.g., the action may include logging an event). For example, it is possible to determine how many commercials the Coca Cola Company in the Chicago market aired in the month of June. This information is valuable to competitors such as Pepsi. Thus, any company that developed a system as described above could also expect to generate revenue from competitive market research data that it gathers.

Advertisers often wish to ensure that they receive the advertising time that was purchased. To do so, they often hire commercial verification services to verify that the advertisement or commercial did indeed run at the expected time. To do so, currently deployed systems by Nielsen and CMR embedded active signals in the advertisement prior to the broadcast. These signals are then detected by remote monitoring facilities that then report back to a central system which commercials were positively identified. See for example U.S. Pat. Nos. 5,629,739 by R. A. Dougherty entitled "Apparatus and method for injecting an ancillary signal into a low energy density portion of a color television frequency spectrum", 4,025,851 by D. E. Haselwood and C. M. Solar entitled "Automatic monitor for programs broadcast", 5,243,423 by J. P. DeJean, D. Lu and R. Weissman, entitled "Spread spectrum digital data transmission over TV video", and 5,450,122 by L. D. Keene entitled "In-station television program encoding and monitoring system and method". (Each of these patents is incorporated herein by reference.) Active systems are usually preferred for advertisement verification because the required recognition accuracy is difficult to achieve with passive systems. The passive monitoring system described herein supports commercial verification.

§4.2.1.2 Exemplary Architectures

Three alternative architectural embodiments in which the first technique may be employed are now described with reference to FIGS. **2**, **3**, and **4**.

FIG. **2** is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work and in which a audience member device **210**, such as a PC for example, receives and renders a work that is consumed by an audience member (user). At some point, the user may wish to perform a work-specific action such as traversing to an associated Web site. Upon initiation of this request, the computer **210** performs the operations **140***a*, **150***a*, **160***a* and **170***a*, such as those shown in FIG. **1**. To reiterate, these operations include a feature extraction operation(s) **140***a*, feature vector lookup or matching operation(s) **150***a* in connection with items or records **112***a* in a work-identification (WID) database **110***a*. If a matching feature vector **114***a* is found, the work-associated information lookup operation(s) **160***a* can use the associated work identifier **116***a* to accessing a work identification-action translation (WIDAT) database **130***a* to retrieve associated information **136***a*, possibly including determining what action should be performed.

As described above, the two databases might be integrated into a single database. However, conceptually, they are described here as separate.

An example illustrating operations that can occur in the first embodiment of FIG. **1**, is now described. Consider a print application, in which say 10,000 advertisements are to be recognized that appear in national newspapers and magazines. If 1 Kbyte is required to store each feature vector then approximately 10 Mbytes of storage will be required for the work identification database **110***a*. Such a size does not represent a serious problem, in either memory or disk space, to present personal computers.

An important issue then becomes recognition rate. While this may be problematic, all the images are two-dimensional—three-dimensional object recognition is not required. Of course, since a low cost camera captures the printed advertisement, there may be a number of geometric distortions that might be introduced together with noise. Nevertheless, the application is sufficiently constrained that adequate recognition rates should be achievable with current state-of-the-art computer vision algorithms. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search",

Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, Jan. 15, 1999. See also, P. N. Yianilos "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. Estimates of the size of the WIDAT database **130***a* depend on what associated information (recall fields **136**) is stored. If, for example, only a URL address is needed, about 20 characters can typically represent most URLs. Thus, the size of the WIDAT database **130***a* would be less than 1 Mbyte.

The configuration just described with reference to FIG. **2** places all of the processing and data on each user's local machine **210**. A number of alternative embodiments, in which some or all of the storage and processing requirements are performed remotely, will be described shortly.

As new works are created and made publicly available, the databases residing on a user's local computer become obsolete. Just as the database provider **240** must continually update the databases in order to remain current, there is also a need to update local databases on devices at audience member premises. This update process can be performed over the Internet **230** in a manner very similar to how software is currently upgraded. It is not necessary to download an entirely new database although this is an option. Rather, only the changes need to be transmitted. During this update process, the user's computer **210** might also transmit information to a central monitoring center **240** informing it of which advertisements the computer user has queried. This type of information is valuable to both advertisers and publishers. Of course, care must be taken to ensure the privacy of individual users of the system. However, it is not necessary to know the identity of individual users for the system to work.

FIG. **3** is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work. Although the WIDAT database can be quite small, as illustrated in the exemplary embodiment described above with respect to FIG. **2**, there is still the problem of keeping this database current. While periodic updates of the local databases may be acceptable, they become unnecessary if the WIDAT database **130***b* is at a remote location **340**. In this arrangement, illustrated in FIG. **3**, after the local computer **310** identifies the work, it sends a query to the remote WIDAT database **130***b*. The query may contain the work identifier. The remote site **340** may then return the associated information **136**. Although the remote WIDAT database **130***b* needs to be updated by the database provider, this can be done very frequently without the need for communicating the updates to the local computers **310**.

The second embodiment is most similar to active systems in which an embedded signal is extracted and decoded and the identifier is used to interrogate a central database. Consequently it has many of the advantages of such systems, while avoiding the need to insert signals into all works. One such advantage, is that the database provider receives real-time information relating to users' access patterns.

The WIDAT database **130***b* might physically reside at more than one location. In such a case, some requests will go to one site, and other requests will go to another. In this way, overloading of a single site by too many users can be avoided. Other load balancing techniques are also applicable.

FIG. **4** is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work. Recall that the WIDAT database may be small relative to that work identification database (WID). As the size of the work recognition (WID) database increases, the foregoing embodiments may become impractical. Consider, for example, a music application in which it is desired to identify 100,000 song titles. If it is again assumed that a 1 Kbyte vector can uniquely represent each song, then on the order of 100 Mbytes is now needed. This size is comparable to large application programs such as Microsoft's Office 2000 suite. Although this still does not represent an inordinate amount of disk space, if this data needs to reside in memory at all times, then very few present machines will have adequate resources. Clearly, at some point, the proposed architectures scales to a point where requirements become impractical. In this case, a further modification to the architecture is possible.

Since the storage and searching of the work-identifier (WID) database require the most computation and storage, it may be more economical to perform these actions remotely. Thus, for example, if a user is playing an MP3 music file and wants to go to a corresponding website, the MP3 file is passed to an operation that determines one or more feature vectors. In the third embodiment, instead of performing the matching locally **410**, the one or more vectors are transmitted to a central site **440** at which is stored the WID and WIDAT databases **110***c* and **130***c* together with sufficiently powerful computers to resolve this request and those of other computer users. This configuration is illustrated in FIG. **4**. Similarly, if a user is playing an MPEG or other video file and wants to initiate a work-related action, the video file is passed to an operation **140***c* that extracts one or more feature vectors. The entire video file need not be processed. Rather, it may be sufficient to process only those frames in the temporal vicinity to the users request, i.e., to process the current frame and or some number of frames before and after the current frame, e.g. perhaps 100 frames in all. The extracted feature vector or feature vectors can then be transmitted to a central site **440** which can resolve the request.

After successfully matching the feature vector, the central site **440** can provide the user with information directly, or can direct the user to another Web site that contains the information the user wants. In cases where the recognition is ambiguous, the central site **440** might return information identifying one of several possible matches and allow the user to select the intended one.

The third embodiment is particularly attractive if the cost of extracting the feature vector is small. In this case, it becomes economical to have feature vector extraction **140c** in digital set-top-boxes and in video recorders **410**. The latter may be especially useful for the new generation of consumer digital video recorders such as those manufactured by TIVO and Replay TV. These devices already have access to the Internet via a phone line. Thus, when someone watching a recorded movie from television reacts to an advertisement, the video recorder would extract one or more feature vectors and transmit them to a central site **440**. This site **440** would determine if a match existed between the query vector and the database of pre-stored vectors **110c**. If a match is found, the central server **440** would transmit the associated information, which might include a Web site address or an 800 number for more traditional ordering, back to the audience user device **410**. Of course, a consumer device **410** such as a digital video recorder might also store personal information of the owner to facilitate online e-commerce. Such a device **410** could store the owner's name, address, and credit card information and automatically transmit them to an on-line store to complete a purchase. Very little user interaction other than to authorize the purchase might be needed. This type of purchasing may be very convenient to consumers.

Another advantage of the third embodiment is that it obviates the need to update local databases while, at the same time, the centrally maintained databases can be kept current with very frequent updating.

§4.2.2 Embodiments in which Work is Recognized Based on Extra-Work Information

Operations related to this embodiment are described in §4.2.2.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.2.2.

If the cost of extracting a feature vector is too large, then the cost of deploying any of the embodiments described in §4.2.1 above may be prohibitive. This is particularly likely in very cost sensitive consumer products, including set-top-boxes and next generation digital VCRs. Acknowledging this fact, a different technique, one that is particularly well suited for broadcasted media such as television and radio as well as to content published in magazines and newspapers, is now described. This technique relies on the fact that a work need not be identified by a feature vector extracted from the work (which is an example of "intra-work information"), but can also be identified by when and where it is published or broadcast (which are examples of "extra-work information")

An example serves to illustrate this point. Consider the scenario in which a viewer sees a television commercial and responds to it. The embodiments described in §4.2.1 above required the user device (e.g., a computer or set-top-box) **210/310/410** to extract a feature vector. Such an extracted vector was attempted to be matched to another feature vector(s), either locally, or at a remote site. In the embodiments using a remote site, if the central site is monitoring all television broadcasts, then the user's query does not need to include the feature vector. Instead, the query simply needs to identify the time, geographic location and the station that the viewer is watching. A central site can then determine which advertisement was airing at that moment and, once again, return the associated information. The same is true for radio broadcasts. Moreover, magazines and newspapers can also be handled in this manner. Here the query might include the name of the magazine, the month of publication and the page number.

§4.2.2.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. **5** is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work. As shown, a query work-identification (QWID) information storage **510** may include a number of items or records **512**. Each item or record **512** may associate extra-work information **514**, related to the work, with a, preferably unique, work identifier **516**. The query work-identification (QWID) information storage **510** may be generated by a database generation operation(s) **520**.

Further, work identifier-action information (WIDAT) storage **530** may include a number of items or records **532**. Each item or record **532** may associate a, preferably unique, work identifier **534** with associated information **536**, such as an action for example. The work identifier-action (WIDAT) information storage **530** may be generated by a database generation operation(s) **538** which may, for example, accept manual entries.

As can be appreciated from the foregoing, the query work-information (QWID) storage **510** records **512** and the work identification-action (WIDAT) storage **530** records **532** can be combined into a single record.

The extra-work information aggregation (e.g., query generation) operation(s) **540** can accept a information related to a work, such as the time of a user request or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and generate a query from such extra-work information.

The query including the extra-work information can be used by a lookup operation(s) **550** to search for a "matching" set of information **514**. If a match, or a match within a predetermined threshold is determined, then the associated work identifier **516** is read.

The read work identifier can then be used by a work-associated information lookup operation(s) **560** to retrieve associated information, such as an action, **536** associated with the work identifier. Such information **536** can then be passed to action initiation operation(s) **570** which can perform some action based on the associated information **536**.

If the extra-work information of a work is known (in advance), generating the query work identifier (QWID) information **510** is straight-forward. If this were always the case, an intra-work information-based recognition operation would not be needed. However, very often this is not the case. For example, local television broadcasts typically have discretion to insert local advertising, as well as national advertising. Thus, it often is not possible to know in advance when, on what station, and where a particular advertisement will play.

In such instances, a real-time (e.g., centralized) monitoring facility **580** may be used to (i) extract feature vectors from a work, (ii) determine a work identifier **116** from the extracted features, and (iii) communicate one or more messages **590** in which extra-work information (e.g., time, channel, geographic market) **592** is associated with a work identifier **594**, to operation(s) **520** for generating query work identification (QWID) information **510**.

§4.2.2.1.1 Exemplary Extra-Work INFORMATION

In the context of national broadcasts, geographic information may be needed to distinguish between, for example, the ABC television broadcast in Los Angeles and that in New York. While both locations broadcast ABC's programming, this programming airs at different times on the East and West coasts of America. More importantly, the local network affiliates that air ABC's shows have discretion to sell local advertising as well as a responsibility to broadcast the national commercials that ABC sells. In short, the works broadcast by

ABC in Los Angeles can be different from that in other geographic locations. Geographic information is therefore useful to distinguish between the different television markets. In some circumstances, geographic information may not be necessary, especially in parts of the world with highly regulated and centralized broadcasting in which there are not regional differences.

§4.2.2.1.2 Exemplary Techniques for Generating Databases

FIG. **5** illustrates a third database **510** referred to as the query to work identification (QWID) database. This database **510** maps the query (e.g., in the form of time, location and channel information) into a unique ID that identifies the perceived work. The QWID **510** and WIDAT **530** databases might not be separate, but for clarity will be considered so. After retrieving the unique work identifier **512** from the QWID database **510**, the identifier can be used to access the WIDAT database **530**. This is discussed in more detail later.

As introduced above, although it appears that this architecture does not require a recognition facility, such a facility may be needed. The feature extraction operation(s) **140***d*, as well as the work identification operation(s) **150***d* and other databases **110***d*, may be moved to one or more remote sites **580**.

Although TV Guide and other companies provide detailed information regarding what will be broadcast when, these scheduling guides do not have any information regarding what advertisements will air when. In many cases, this information is unknown until a day or so before the broadcast. Even then, the time slots that a broadcaster sells to an advertiser only provide a time range, e.g. 12 pm to 3 pm. Thus it is unlikely that all commercials and aired programming can be determined from TV schedules and other sources prior to transmission. Further, occasionally programming schedules are altered unexpectedly due to live broadcasts that overrun their time slots. This is common in sports events and awards shows. Another example of interrupts to scheduled programming occurs when a particularly important news event occurs.

During transmission, it may therefore be necessary for a central site **580** to determine what work is being broadcast and to update its and/or other's database **520** accordingly based on the work identified **594** and relevant extra-work information **592**. There are a variety of ways that this can be accomplished.

First, it may be economically feasible to manually monitor all television stations that are of interest, and manually update the database with information regarding the work being monitored. In fact, Nielsen used such procedures in the early 1960's for the company to tabulate competitive market data. More than one person can be employed to watch the same channel in order to reduce the error rate. It should be noted that the recent ruling by the FCC that satellite broadcasters such as DirecTV, DishTV and EchoStar can carry local stations significantly reduces the cost of monitoring many geographic markets. Currently, DirecTV, for example, carries the four main local stations in each of the 35 largest markets. Thus, these 4.times.35=140 channels can all be monitored from a single site **580**. This site would be provided with satellite receivers to obtain the television channels.

Unfortunately, however, humans are error prone and the monitoring of many different stations from many different geographic locations can be expensive. In order to automate the recognition process, a central site **580** could employ a computer-based system to perform automatic recognition. Because the recognition is centralized, only one or a few sites are needed. This is in comparison with the first architecture we described in which a complete recognition system was required in every user's home or premise. This centralization makes it more economic to employ more expensive computers, perhaps even special purpose hardware, and more sophisticated software algorithms. When video frames or clips cannot be identified or are considered ambiguous, this video can be quickly passed to human viewers to identify. Further, it should be possible for the automated recognition system to use additional information such as television schedules, time of day, etc in order to improve its recognition rate.

§4.2.2.1.2 Exemplary Techniques for Generating Queries Based on Extra-Work Information

At the audience member (user) premises, all that is needed is for the device to send a query to a database-server with information that includes extra-work information, such as geographic location, time and channel. Usually, this extra-work information would be transmitted in real-time, while the work (e.g., an advertisement) is being broadcast. However, this is not necessary. If the television does not have access to the Internet, and most TVs do not yet, then an audience member (user) may simply remember or record which channel he or she was viewing at what time. In fact, the user device could store this information for later retrieval by the user. At a convenient later time, the user might access the Internet using a home PC. At this time, he or she can query the database by entering this extra-work information (e.g., together with geographic information) into an application program or a web browser plug-in.

Another possibility is allowing an audience member (user), at the time he or she is consuming (e.g., viewing, reading, listening to, etc.) the work, to enter query information into a handheld personal digital assistant ("PDA") such as a Palm Pilot, so as not to forget it. This information can then be manually transferred to a device connected to a network, or the information can be transferred automatically using, for example, infrared communications or via a physical link such as a cradle. Recently, PDAs also have some wireless networking capabilities built in, and thus might support direct access to the information desired. Further, software is available that allows a Palm Pilot or other PDA to function as a TV remote control device. As such, the PDA already knows the time of day and channel being viewed. It also probably knows the location of the audience member, since most PDA users include their own name and address in the PDA's phonebook and identify it as their own. Thus, with one or a few clicks, an audience member PDA user could bookmark the television content he or she is viewing. If the PDA is networked, then the PDA can, itself, retrieve the associated information immediately. Otherwise, the PDA can transfer this bookmarked data to a networked device, which can then provide access to the central database.

§4.2.2.2 Exemplary Architectures

FIG. **6** is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work. As shown, an extra-work information aggregation operation **540***a* may be effected on a device **610**, such as a PC, at the audience member (user) premises. The various databases **510***a*, **530***a*, and **110***e*, as well as the database generation operation(s) **520***a*/**538***a*, the lookup operation(s) **550***a* and the work-associated information lookup operation(s) **560***a* may be provided at one or more centralized monitoring and query resolution centers **640**.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work. This fifth embodiment is similar to the fourth embodiment illustrated in FIG. 6 but here, the monitoring center **740***a* and query resolution center **740***b* are separate.

These embodiments have many advantages for television and radio broadcasters who desire to provide Internet links or other action. First, the audience member (user) equipment, whether it is a computer, set-top-box, television, radio, remote control, personal digital assistant (pda), cell phone or other device, does not need to perform any processing of the received signal. As such, there is almost no cost involved to equipment manufacturers.

These last embodiments have some similarity with services such as those provided by the companies Real Names of Redwood City, Calif., America Online ("AOL") and especially iTag from Xenote. The popular press has reported on the difficulties associated with assigning domain names. The simplest of these problems is that almost all the one-word names in the ".com" category have been used. Consequently, domain names can often be difficult to remember. To alleviate this problem, RealNames and AOL provide alternative, proprietary name spaces (AOL calls these keywords). For a fee, a company may register a name with these companies. Thus, rather than type the URL http://www.bell-labs.com, the simple keyword "bell" might be sufficient to access the same Web site. These capabilities are convenient to users. However, these systems are very different from the fourth and fifth embodiments described. First, and foremost, these systems are not designed to identify content. Rather, they are simply alternative network address translation systems based on easily remembered mnemonics which are sold to interested companies. As such, the user is still expected to type in an address, but this address is easier to remember than the equivalent URL. In contrast, while a user may manually enter the information describing the work, the preferred embodiment is for the computer, set-top-box or other device to automatically generate this information. Further, the mapping of keywords to network addresses is an arbitrary mapping maintained by AOL or Real Names. For example, the keyword "bell" might just as reasonably point to the Web site for Philadelphia's Liberty Bell as to Lucent's Bell Labs. In contrast, the query used in the fourth and fifth embodiments is designed to contain all the necessary data to identify the work, e.g. the time, place and television channel during which the work was broadcast. There is nothing arbitrary about this mapping. It should also be pointed out that the proposed system is dynamic—the same work, e.g. a commercial, potentially has an infinite number of addresses depending on when and where it is broadcast. If an advertisement airs 100,000 unique times, then there are 100,000 different queries that uniquely identify it. Moreover, the exemplary query includes naturally occurring information such as time, place, channel or page number. This is not the case for AOL or RealNames, which typically assigns one or more static keywords to the address of a Web site.

Xenote's iTag system is designed to identify radio broadcasts and uses a query similar to that which may be used in the fourth and fifth embodiments, i.e. time and station information. However, the work identification information is not dynamically constructed but is instead based on detailed program scheduling that radio stations must provide it. As such, it suffers from potential errors in scheduling and requires the detailed cooperation of broadcasters. While the fourth and fifth embodiments might choose to use program scheduling information and other ancillary information to aid in the recognition process, they do not exclusively rely on this. The concept of resolving a site name by recognizing the content is absent from the above systems.

§4.2.3 Exemplary Apparatus for Audience Member (User) Premise Device

While personal computers may be the primary computational device at a user's location, it is not essential to use a PC. This is especially true of the embodiments depicted in FIGS. 6 and 7, which do not require the content, e.g. video signal, to be processed. Instead, only a unique set of identification parameters such as time, location and channel are provided to identify the perceived Work. Many forms of devices can therefore take advantage of this configuration.

As previously noted, personal digital assistants (PDAs) can be used to record the identification information. This information can then be transferred to a device with a network communication such as a PC. However, increasingly, PDAs will already have wireless network communication capabilities built-in, as with the Palm VII PDA. These devices will allow immediate communication with the query resolution center and all information will be downloaded to them or they can participate in facilitating an e-commerce transaction. Similarly, wireless telephones are increasingly offering web-enabled capabilities. Consequently, wireless phones could be programmed to act as a user interface.

New devices can also be envisaged, including a universal remote control for home entertainment systems with a LCD or other graphical display and a network connection. This connection may be wireless or the remote control might have a phone jack that allows it to be plugged directly into an existing phone line. As home networks begin to be deployed, such devices can be expected to communicate via an inexpensive interface to the home network and from there to access the Internet.

In many homes, it is not uncommon for a computer and television to be used simultaneously, perhaps in the same room. A person watching television could install a web browser plug-in or applet that would ask the user to identify his location and the station being watched. Then, periodically, every 20 seconds for example, the plug-in would update a list of web addresses that are relevant to the television programs being watched, including the commercials. The audience member would then simply click on the web address of interest to obtain further information. This has the advantage that the viewer does not have to guess the relevant address associated with a commercial and, in fact, can be directed to a more specialized address, such as www.fordvehicles.com/ibv/tausrus2kflash/flash.html, rather than the generic www.ford.com site. Of course, this applet or plug-in could also provide the database entity with information regarding what is being accessed from where and at what time. This information, as noted earlier, is valuable to advertisers and broadcasters. For PC's that have infra-red communication capabilities, it is straightforward to either control the home entertainment center from the PC or for the PC to decode the signals from a conventional remote control. Thus, as a user changes channels, the PC is able to automatically track the channel changes.

Recording devices such as analog VCRs and newer digital recording devices can also be exploited in the embodiments depicted in FIGS. 6 and 7, especially if device also record the channel and time information for the recorded content. When a user initiates a query, the recorded time and channel, rather than the current time and channel, then form part of the identification information.

Digital set-top-boxes are also expected to exploit the capabilities described herein. In particular, such devices will have two-way communication capabilities and may even include cable modem capabilities of course, the two-way communication need not be over a television cable. For example, satellite set-top-boxes provide up-link communications via a telephone connection. Clearly, such devices provide a convenient location to enable the services described herein. Moreover, such services can be provided as part of the OpenCable and DOCSIS (data over cable service interface specification) initiatives.

§4.2.4 Information Retrieval Using Features Extracted from Audio and/or Video Works

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable program for providing information about an audio file or (a video file) played on a device. Such embodiments might (a) extract features from the audio (or video) file, (b) communicate the features to a database, and (c) receive the information about the audio (or video) file from the database. In some embodiments consistent with the present invention, the act of extracting the features is performed by a microprocessor of the device, and/or a digital signal processor of the device. The received information might be rendered on an output (e.g., a monitor, a speaker, etc.) of the device. The received information might be stored (e.g., persistently) locally on the device. The information might be stored on a disk, or non-volatile memory.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§4.3 OPERATIONAL EXAMPLES

An example illustrating operations of an exemplary embodiment of the present invention, that uses intra-work information to identify the work, is provided in §4.3.1. Then, an example illustrating operations of an exemplary embodiment of the present invention, that uses extra-work information to identify the work, is provided in §4.3.2.

§4.3.1 Operational Example where Intra-Work Information is Used to Identify the Work

A generic system for monitoring television commercials is now described. Obviously, the basic ideas extend beyond this specific application.

The process of recognition usually begins by recognizing the start of a commercial. This can be accomplished by looking for black video frames before and after a commercial. If a number of black frames are detected and subsequently a similar number are detected 30 seconds later, then there is a good chance that a commercial has aired and that others will follow. It is also well known than the average sound volume during commercials is higher than that for television shows and this too can be used as an indicator of a commercial. Other methods can also be used. The need to recognize the beginning of a commercial is not essential. However, without this stage, all television programming must be assumed to be commercials. As such, all video frames must be analyzed. The advantage of determining the presence of a commercial is that less video content must be processed. Since the percentage of advertising time is relatively small, this can lead to considerable savings. For example, commercials can be buffered and then subsequently processed while the television show is being broadcast. This reduces the real-time requirements of a system at the expense of buffering, which requires memory or disk space. Of course, for the applications envisioned herein, a real-time response to a user requires real-time processing.

Once it is determined that an advertisement is being broadcast, it is necessary to analyze the video frames. Typically, a compact representation of each frame is extracted. This vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n.times.n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, Fourier-Mellin, wavelet and or discrete cosine transforms. It might involve principal component analysis or any combination thereof. The recognition literature contains many different representations. For block-based methods, the n.times.n blocks may be located at pseudo-random locations in each frame or might have a specific structure, e.g. a complete tiling of the frame. The feature vector might then be composed of the pixels in each block or some property of each block, e.g. the average intensity or a Fourier or other decomposition of the block. The object of the vector extraction stage is to obtain a more concise representation of the frame. Each frame is initially composed of 480.times.720 pixels which is equivalent to 345,600 bytes, assuming one byte per pixel. In comparison, the feature vector might only consist of 1 Kbyte of data. For example, if each frame is completely tiled with 16.times.16 blocks, then the number of blocks per frame is 345,600/256=1350. If the average intensity of each block constitutes the feature vector, then the feature vector consists of 1350 bytes, assuming 8-bit precision for the average intensity values. Alternatively, 100 16.times.16 blocks can be pseudo-randomly located on each frame of the video. For each of these 100 blocks, the first 10 DCT coefficients can be determined. The feature vector then consists of the 100.times.10=1000 DCT coefficients. Many other variations are also possible. In many media applications, the content possesses strong temporal and spatial correlations. If necessary, these correlations can be eliminated or substantially reduced by pre-processing the content with a whitening filter.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e. horizontal and or vertical translation, or may undergo lossy compression such as MPEG-2. It is advantageous, though not essential, that these and other processes do not adversely affect the extracted vectors.

Each frame's feature vector is then compared with a database of known feature vectors. These known vectors have previously been entered into a content recognition database together with a unique identifier. If a frame's vector matches a known vector, then the commercial is recognized. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when a frame's vector is not matched to the database even though the advertisement is present in the database. There are several reasons why a frame's feature vector may fail to match. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible. Finally, there is the case where the observed commercial is not yet present in the database. In this case, it is necessary to store the commercial and pass it (e.g., to a person) for identification and subsequent entry in the database.

It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search is often not possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used, including clustering techniques. See, e.g., the Duda and Hart reference. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match.

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. However, in current advertisement monitoring applications there is no discussion of efficient search methods. Thus, a linear search of all N entries may be performed, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this can be computationally expensive. Consider a situation in which one out of 100,000 possible commercials is to be identified. Each 30-second commercial consists of 900 video frames. If all 900 frames are stored in the database, then N=90,000,000. Even if only every 10.sup.th video frame is stored in the database, its size is still nine million. While databases of this size are now common, they rely of efficient search to access entries, i.e., they do not perform a linear search. A binary search of a 90,000,000-item database requires less than 20 comparisons. In contrast, a linear search will require an average of 45,000,000!

With 9 million entries, if each vector is 1 Kbyte, then the storage requirement is 9 Gigabytes. Disk drives with this capacity are extremely cheap at this time. However, if the database must reside in memory due to real-time requirements, then this still represents a substantial memory requirement by today's standards. One reason that the data may need to be stored in memory is because of the real-time requirements of the database. If 10 channels are being simultaneously monitored within each of 50 geographic areas, then there will be 15,000 queries per second to the content recognition database, assuming each and every frame is analyzed. This query rate is low. However, if a linear search is performed then 675 billion comparisons per second will be required. This is an extremely high computational rate by today's standards. Even if only key frames are analyzed, this is unlikely to reduce the computational rate by more than an order of magnitude.

If an advertisement is not recognized, then typically, the remote monitoring system will compress the video and transmit it back to a central office. Here, the clip is identified and added to the database and the remote recognition sites are subsequently updated. Identification and annotation may be performed manually. However, automatic annotation is also possible using optical character recognition software on each frame of video, speech recognition software, close captioning information and other information sources. As these methods improve in accuracy, it is expected that they will replace manual identification and annotation.

The recognition system described can be considered to be a form of nearest neighbor search in a high dimensional feature space. This problem has been very well studied and is known to be very difficult as the dimensionality of the vectors increases. A number of possible data structures are applicable including kd-trees and vantage point trees. These data structures and associated search algorithms organize a N-point dataset (N=90,000,000 in out previous example) so that sub-linear time searches can be performed on average. However, worst-case search times can be considerably longer. Recently, Yianilos proposed an excluded middle vantage point forest for nearest neighbor search. See, e.g., the Yianilos reference. This data structure guarantees sub-linear worst-case search times, but where the search is now for a nearest neighbor within a fixed radius, .tau. The fixed radius search means that if the database contains a vector that is within .tau. of the query, then there is a match. Otherwise, no match is found. In contrast, traditional vantage point trees will always return a nearest neighbor, even if the distance between the neighbor and the query is very large. In these cases, if the distance between the query and the nearest neighbor exceeds a threshold, then they are considered not to match. This is precisely what the excluded middle vantage point forest implicitly does.

Using an excluded middle vantage point forest, will allow accurate real-time recognition of 100,000 broadcasted advertisements. This entails constructing an excluded middle vantage point forest based on feature vectors extracted from say 90,000,000 frames of video. Of course, using some form of pre-filtering that eliminates a large number of redundant frames or frames that are not considered to be good unique identifiers can reduce this number. One such pre-filter would be to only examine the I-frames used when applying MPEG compression. However, this is unlikely to reduce the work identification database (WID) size by more than one order of magnitude. Assuming 10 channels are monitored in each of 50 geographic regions, then the query rate is 15,000=10.times.50.times.30 queries per second.

§4.3.2 Operational Example where Extra-Work Information is Used to Identify the Work

FIG. **8** depicts a satellite television broadcast system **800**, though cable and traditional broadcast modes are also applicable. Block **810** represents audience members (users) watching a TV channel in their home, which also has a connection **812** to the Internet **820**. Other networks are also possible. The satellite broadcasts are also being monitored by one or more television monitoring centers **840***a*. These centers **840***a* may monitor all or a subset of the television channels being broadcast. They are not restricted to monitoring satellite TV broadcasts but may also monitor cable and traditional terrestrial broadcasts. The primary purpose of these monitoring centers **840***a* is to identify the works being broadcasted. Of particular interest are television advertisements. However, other works, or portions thereof, may also be identified. Each time a new segment of a work is identified, the monitoring system or systems **840***a* update one or more database centers **840***b*, informing them of the time, place, channel and identity of the identified segment. The segment may be a complete thirty second commercial or, more likely, updates will occur more frequently, perhaps at a rate of 1 update per second per channel per geographic location. The database center **840***b* updates its database so that queries can be efficiently responded to in sub-linear time.

The database centers **840***b* can use traditional database technology. In general, the query search initiated by an audience member is not a nearest neighbor search but can be a classical textual search procedure such as a binary search. The nearest neighbor search is appropriate for the monitoring sub-system **840***a*. The database centers **840***b* are continually updated as each new advertisement, television show or portion thereof is recognized. Standard updating algorithms can be used. However, random new entries to the database are unlikely. Rather, each new entry, or set of entries, denotes a new time segment that is later than all previously inserted items. As such, each new entry can be appended to the end of the database while still maintaining an ordered data structure that is amenable to binary and other efficient search techniques. If two entries have the same time in their time field, items can be sorted based on secondary fields such as the channel and geographic location, as depicted in FIG. **9**. Since the number of such entries will be relatively small compared with the entire database, it may be sufficient to simply create a linear linked list of such entries, as depicted in FIG. **9**. Of course, the size of the database is constantly increasing. As such, it may become necessary to have several levels of storage and caching. Given the envisaged application, most user queries will be for recent entries. Thus, the database may keep the last hours worth of entries in memory. If there is one entry per second for each of 100 channels in 100 geographic locations, this would correspond to 3600.times.100.times.100=36,000,000 entries which is easily accommodated in main memory. Entries that are older than one hour may be stored on disk and entries older than one week may be archived (e.g., backed up on tape) for example. The entries to this database can include time, location and channel information together with a unique identifier that is provided by the monitoring system. Of course, additional fields for each entry are also possible.

When a user query is received, the time, channel and geographic information are used to retrieve the corresponding unique identifier that is then used to access a second database that contains information associated with the identified work.

An entry **1000** in this second database is depicted in FIG. **10**, which shows that associated with the unique identifier **1010**, the name of a product **1020**, a product category **1030**, the manufacturer **1040** and the commercial's associated web site **1050**. Many other data fields **1060** are also possible. Such additional fields may include fields that indicate what action should be taken on behalf of the requesting user. Example actions include simply redirecting a request to an associated Web site, or initiating an e-commerce transaction or providing an associated telephone number that may be automatically dialed if the querying device is a cell phone or displaying additional information to the user. This database is likely to be updated much less frequently, perhaps only as often as once or twice a day, as batches of new advertisements are added to the system. Alternatively, it might be updated as each new advertisement is added to the system.

An audience member (user) **810** watching a television commercial for example may react to the advertisement by initiating a query to the database center **840***b*. The device whereby the user initiates the query might be a television or set-top-box remote control, or a computer or a wireless PDA or a (WAP-enabled) cell phone or a specialized device. Typically, the query will occur during the airing of the commercial or a shortly thereafter. However, the time between the broadcasting of the advertisement and the time of the associated query is not critical and can, in some instances be much longer. For example, the audience member might bookmark the query information in a device such as a PDA or a specialized device similar to those developed by Xenote for their Itag radio linking. Later, the audience member may transmit the query to the database center **840***b*. This might happen hours or even days later.

The query contains information that the database center **840***b* uses to identify the work being viewed. This information might include the time and place where the audience member was, together with the channel being viewed. Other identifying information is also possible. The query may also contain additional information that may be used to facilitate the user's transaction and will include the return address of the user. For example, if the user is intending to order a pizza after seeing a Pizza Hut advertisement, the query may also contain personal information including his or her identity, street address and credit card information.

When the database center **840***b* receives a query, data in the query is used to identify the work and associated information. A number of possible actions are possible at this point. First, the database center **840***b* may simply function as a form of proxy server, mapping the audience member's initial query into a web address associated with the advertisement. In this case, the audience member will be sent to the corresponding Web site. The database center **840***b* may also send additional data included in the initial query to this Web site **850** in order to facilitate an e-commerce transaction between the audience member and the advertiser. In some cases, this transaction will not be direct, but may be indirect via a dealer or third party application service provider. Thus, for example, though an advertisement by Ford Motor Company may air nationally, viewers may be directed to different Web sites for Ford dealerships depending on both the audience member's and the dealerships' geographic locations. In other cases, advertisers may have contracted with the database center **840***b* to provide e-commerce capabilities. This latter arrangement has the potential to reduce the amount of traffic directed over the public Internet, restricting it, instead to a private network associated with the owner of the database center.

If the audience member (user) is not watching live television but is instead watching a taped and therefore time-shifted copy, then additional processes are needed. For the new generation of digital video recorders, irrespective of the recording media (tape or disk), it is likely to be very easy to include information identifying the location of the recorder, as well as the time and channel recorded. Location information can be provided to the recorder during the setup and installation process, for example. Digital video recorders, such as those currently manufactured by TIVO of Alviso, Calif. or Replay TV of Santa Clara, Calif. have a network connection via telephone, which can then send the query of an audience member to the database center **840***b* using the recorded rather than the current information.

In cases where query information has not been recorded, it is still possible to initiate a successful query. However, in this case, it may be necessary to extract the feature vector from the work of interest and send this information to the monitoring center **840***a* where the feature vector can be identified. This form of query is computationally more expensive but the relative number of such queries compared to those sent to the database centers **840***b* is expected to be small. It should also be noted that the physical separation of the monitoring and database centers, depicted in FIGS. **6** and **7**, is not crucial to operation of the system and simply serves to more clearly separate the different functionality present in the overall system configuration.

Although the implementation architectures described above focus on the television media, it is apparent that the present invention is applicable to audio, print and other media.

§4.4 CONCLUSIONS

None of the embodiments of the invention require modification to the work or content, i.e., no active signal is embedded. Consequently, there is no change to the production processes. More importantly, from a user perspective, deployment of this system need not suffer from poor initial coverage. Provided the database is sufficiently comprehensive, early adopters will have comprehensive coverage immediately. Thus, there is less risk that the consumer will perceive that the initial performance of the deployed system is poor. Further, the present invention permits statistics to be gathered that measure users' responses to content. This information is expected to be very useful to advertisers and publishers and broadcasters.

What is claimed is:

1. A computer-implemented method comprising:

(a) maintaining, by a computer system including at least one computer, a database comprising:

(1) first electronic data related to identification of one or more reference electronic works; and

(2) second electronic data related to action information comprising an action to perform corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, extracted features of a first electronic work;

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action information corresponding to the identified first electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action information with the identified first electronic work.

2. The computer-implemented method of claim 1, wherein the step of obtaining comprises receiving the first electronic work and extracting the extracted features from the first electronic work.

3. The computer-implemented method of claim 2, wherein the first electronic work is received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

4. The computer-implemented method of claim 1, wherein the step of obtaining comprises receiving the extracted features.

5. The computer-implemented method of claim 4, wherein the extracted features are received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

6. The computer-implemented method of claim 1, wherein the step of obtaining comprises receiving the first electronic work and the extracted features of the first electronic work.

7. The computer-implemented method of claim 6, wherein the first electronic work and the extracted features of the first electronic work are received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

8. The computer-implemented method of claim 1, wherein at least one of the first electronic work or the extracted features of the first electronic work is obtained from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device that stores commercial transaction data, and wherein the action information comprises a commercial transaction, and the method further comprises:

electronically performing the commercial transaction using the stored commercial transaction data.

9. The computer-implemented method of claim 1, wherein at least one of the first electronic work or the extracted features of the first electronic work is obtained from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device, and wherein the action information comprises commercial transaction data, and the method further comprises:

electronically performing the commercial transaction through the at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device using the commercial transaction data.

10. The computer-implemented method of claim 8, wherein the stored commercial transaction data comprises at least one of a purchaser's name, a purchaser's address, or credit card information.

11. The computer-implemented method of claim 1, wherein the action comprises at least one of directing a user to a website related to the first electronic work, initiating an e-commerce transaction, or dialing a phone number.

12. The computer-implemented method of claim 1, wherein the action comprises providing and/or displaying additional information in association with the first electronic work.

13. A computer-implemented method comprising:

(a) maintaining, by a computer system, one or more databases comprising:

(1) first electronic data comprising a first digitally created compact electronic representation of one or more reference electronic works; and

(2) second electronic data related to an action, the action comprising providing and/or displaying an advertisement corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, a second digitally created compact electronic representation of a first electronic work;

(c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action corresponding to the matching reference electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action with the first electronic work.

14. The computer-implemented method of claim 13, wherein the action further comprises providing a link to a site on the World Wide Web associated with the advertisement.

15. The computer-implemented method of claim 13, wherein the action further comprises electronically registering a user with at least one of a service or a product related to the advertisement.

16. The computer-implemented method of claim 13, wherein the action further comprises electronically providing at least one of a coupon or a certificate related to the advertisement.

17. The computer-implemented method of claim 13, wherein the action further comprises automatically dialing a telephone number associated with the advertisement.

18. The computer-implemented method of claim 13, wherein the action further comprises collecting competitive market research data related to the advertisement.

19. The computer-implemented method of claim 13, wherein the action further comprises purchasing a product or service related to the advertisement.

20. The computer-implemented method of claim 13, wherein the action further comprises allowing a user to interact with a broadcast related to the advertisement.

21. The computer-implemented method of claim 1, further comprising the step of transmitting, by the computer system, the determined action as associated with the first electronic work.

22. The computer-implemented method of claim 1, wherein the one or more electronic works comprise at least one of a video work, an audio work, or an image work.

23. The computer-implemented method of claim 1, wherein the method further comprises the additional steps of:

(f) obtaining, by the computer system, second extracted features of a second electronic work;

(g) searching, by the computer system, for an identification of the second electronic work by comparing the second extracted features of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that the second electronic work is not identified based on results of the searching step.

24. The computer-implemented method of claim 13, wherein the method further comprises the additional steps of:

(f) obtaining, by the computer system, second extracted features of a second electronic work to be identified;

(g) searching, by the computer system, for an identification of the second electronic work by comparing the second extracted features of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that the second electronic work is not identified based on results of the searching step.

25. A computer-implemented method comprising the steps of:

(a) maintaining, by a computer system, one or more databases comprising:

(1) first electronic data comprising a first digitally created compact electronic representation comprising an extracted feature vector of one or more reference electronic works; and

(2) second electronic data related to action information, the action information comprising an action to perform corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, a second digitally created compact electronic representation comprising an extracted feature vector of a first electronic work;

(c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation of the first electronic work using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action information corresponding to the matching reference electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action information with the first electronic work.

26. The computer-implemented method of claim 25, wherein the one or more computer readable media have stored thereon further computer instructions for carrying out the additional steps of:

(f) obtaining, by the computer system, a third digitally created compact electronic representation comprising an extracted feature vector of a second electronic work;

(g) searching, by the computer system, for a matching reference electronic work that matches the second electronic work by comparing the third digitally created compact electronic representation of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that a matching reference electronic work that matches the second electronic work does not exist based on results of the searching step.

27. The method of claim 13, wherein the extracted feature is extracted using frequency based decomposition.

28. The computer-implemented method of claim 13, wherein the extracted feature is extracted using principal component analysis.

29. The computer-implemented method of claim 13, wherein the extracted feature is extracted using temporal sequence of feature vectors.

30. The computer-implemented method of claim 13, wherein the first electronic work is obtained from a first user device that is at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device, and wherein the action information comprises a commercial transaction, and the method further comprises:

electronically performing the commercial transaction using commercial transaction data obtained using the first user device.

31. The computer-implemented method of claim 30, wherein the commercial transaction data comprises at least one of a purchaser's name, a purchaser's address, or credit card information.

32. The computer-implemented method of claim 13, wherein the extracted features of the first electronic work are obtained from a first user device that is at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device that stores commercial transaction data, and wherein the action information comprises a commercial transaction, and the method further comprises:

electronically performing the commercial transaction using commercial transaction data obtained using the first user device.

33. The computer-implemented method of claim 32, wherein the commercial transaction data comprises at least one of a purchaser's name, a purchaser's address, or credit card information.

34. The computer-implemented method of claim 13, further comprising the step of transmitting, by the computer system, the determined action information as associated with the first electronic identified work.

35. The computer-implemented method of claim 13, wherein the one or more electronic works comprise at least one of a video work, an audio work, or an image work.

36. The computer-implemented method of claim 25, further comprising the step of transmitting, by the computer system, the determined action information as associated with the first electronic identified work.

37. The computer-implemented method of claim 35, wherein the one or more electronic works comprise at least one of a video work, an audio work, or an image work.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,640,179 B1
APPLICATION NO. : 13/338079
DATED : January 28, 2014
INVENTOR(S) : Ingemar J. Cox

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims

Column 28, lines 18, 20, 23 and 39, for the claim reference numeral "13", each occurrence, should read --1--.

Column 28, line 64, for the claim reference numeral "35" should read --25--.

Signed and Sealed this
Thirteenth Day of May, 2014

Michelle K. Lee

Michelle K. Lee
Deputy Director of the United States Patent and Trademark Office

Google Ex. 1001

US008656441B1

(12) United States Patent
Cox

(10) Patent No.: **US 8,656,441 B1**
(45) Date of Patent: ***Feb. 18, 2014**

(54) **SYSTEM FOR USING EXTRACTED FEATURES FROM AN ELECTRONIC WORK**

(71) Applicant: **Network-1 Security Solutions, Inc.**, New York, NY (US)

(72) Inventor: **Ingemar J. Cox**, London (GB)

(73) Assignee: **Network-1 Technologies, Inc.**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/829,717**

(22) Filed: **Mar. 14, 2013**

Related U.S. Application Data

(63) Continuation of application No. 13/800,573, filed on Mar. 13, 2013, which is a continuation of application No. 13/338,079, filed on Dec. 27, 2011, which is a continuation of application No. 11/977,202, filed on Oct. 23, 2007, now Pat. No. 8,205,237, which is a continuation of application No. 11/445,928, filed on Jun. 2, 2006, now Pat. No. 8,010,988, which is a continuation-in-part of application No. 09/950,972, filed on Sep. 13, 2001, now Pat. No. 7,058,223.

(60) Provisional application No. 60/232,618, filed on Sep. 14, 2000.

(51) **Int. Cl.**
H04N 7/173 (2011.01)

(52) **U.S. Cl.**
USPC **725/115**; 725/110; 725/114; 725/116

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,919,479 A 11/1975 Moon et al.
4,230,990 A 10/1980 Lert, Jr. et al.
(Continued)

FOREIGN PATENT DOCUMENTS

EP 1 354 276 B1 12/2007
EP 1354276 B1 12/2007
(Continued)

OTHER PUBLICATIONS

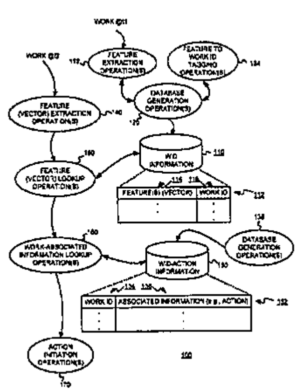
Peter Yianlos, Excluded Middle Vantage Point Forest for Nearest Neighbor Search, Aug. 1, 1999, pp. 1-12.*
(Continued)

Primary Examiner — Cai Chen
(74) *Attorney, Agent, or Firm* — Amster, Rothstein & Ebenstein LLP

(57) **ABSTRACT**

A computer system comprising one or more communications devices; one or more processors operatively connected to the one or more communications devices; and one or more computer readable media operatively connected to the one or more processors and having stored thereon computer instructions for carrying out the steps of (a) maintaining, by the computer system, a database comprising (1) first data related to identification of one or more works; and (2) second data related to information corresponding to each of the one or more works as identified by the first data; (b) obtaining, by the computer system, extracted features of a work to be identified; (c) identifying, by the computer system, the work by comparing the extracted features of the work with the first data in the database using a non-exhaustive neighbor search; (d) determining, by the computer system, the information corresponding to the identified work based on the second data in the database; and (e) associating, by the computer system, the determined information with the identified work.

30 Claims, 10 Drawing Sheets



DEPOSITION EXHIBIT Karypis-5 11/12/15

Google Ex. 1001

Google Ex. 1020

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,450,531	A	5/1984	Kenyon et al.
4,454,594	A	6/1984	Heffron et al.
4,495,526	A	1/1985	Baranoff-Rossine
4,499,601	A	2/1985	Matthews
4,511,917	A	4/1985	Kohler et al.
4,547,804	A	10/1985	Greenberg
4,634,966	A	1/1987	Nakatani et al.
4,639,779	A	1/1987	Greenberg
4,677,455	A	6/1987	Okajima
4,677,466	A	6/1987	Lert, Jr. et al.
4,682,370	A	7/1987	Matthews
4,697,209	A	9/1987	Kiervit et al.
4,739,398	A	4/1988	Thomas et al.
4,776,017	A	10/1988	Fujimoto
4,805,020	A	2/1989	Greenberg
4,843,526	A	6/1989	Price, III
4,843,562	A	6/1989	Kenyon et al.
4,918,730	A	4/1990	Schulze
5,210,820	A	5/1993	Kenyon
5,283,819	A	2/1994	Glick et al.
5,437,050	A	7/1995	Lamb et al.
5,465,353	A	11/1995	Hull et al.
5,481,294	A	1/1996	Thomas et al.
5,504,518	A	4/1996	Ellis et al.
5,550,735	A	8/1996	Slade et al.
5,581,658	A	12/1996	O'Hagan et al.
5,594,934	A	1/1997	Lu et al.
5,629,739	A	5/1997	Dougherty
5,634,012	A	5/1997	Stefik et al.
5,638,443	A	6/1997	Stefik et al.
5,692,213	A	11/1997	Goldberg et al.
5,701,452	A	12/1997	Siefert
5,701,542	A	12/1997	Sasayama
5,706,364	A	1/1998	Kopec et al.
5,724,605	A	3/1998	Wissner
5,745,900	A	4/1998	Burrows
5,748,783	A	5/1998	Rhoads
5,768,426	A	6/1998	Rhoads
5,798,785	A	8/1998	Hendricks et al.
5,822,436	A	10/1998	Rhoads
5,832,119	A	11/1998	Rhoads
5,841,978	A	11/1998	Rhoads
5,850,490	A	12/1998	Johnson
5,862,260	A	1/1999	Rhoads
5,892,536	A	4/1999	Logan et al.
5,918,223	A	6/1999	Blum et al.
5,953,415	A	9/1999	Nielsen
5,963,966	A	10/1999	Mitchell et al.
5,973,723	A	10/1999	DeLuca
5,978,791	A	11/1999	Farber et al.
5,983,171	A	11/1999	Yokoyama et al.
5,983,176	A	11/1999	Hoffert et al.
6,006,256	A	12/1999	Zdepski et al.
6,009,410	A	12/1999	LeMole et al.
6,011,758	A	1/2000	Dockes et al.
6,023,693	A	2/2000	Masuoka et al.
6,026,439	A	2/2000	Chowdhury et al.
6,044,402	A	3/2000	Jacobson et al.
6,052,693	A	4/2000	Smith et al.
6,057,872	A	5/2000	Candelore
6,061,056	A	5/2000	Menard et al.
6,067,369	A	5/2000	Kamei
6,088,455	A	7/2000	Logan et al.
6,088,707	A	7/2000	Bates et al.
6,096,961	A	8/2000	Bruti et al.
6,118,450	A	9/2000	Proehl et al.
6,119,124	A	9/2000	Broder et al.
6,121,530	A	9/2000	Sonoda
6,154,737	A	11/2000	Inaba et al.
6,169,986	B1	1/2001	Bowman et al.
6,173,406	B1	1/2001	Wang et al.
6,195,693	B1	2/2001	Berry et al.
6,229,922	B1	5/2001	Sasakawa et al.
6,233,682	B1	5/2001	Fritsch
6,236,758	B1	5/2001	Sodagar et al.
6,240,409	B1	5/2001	Aiken
6,243,725	B1	6/2001	Hempleman et al.
6,247,133	B1	6/2001	Palage et al.
6,253,193	B1	6/2001	Ginter et al.
6,263,348	B1	7/2001	Kathrow et al.
6,269,275	B1	7/2001	Slade
6,279,010	B1	8/2001	Anderson
6,317,885	B1	11/2001	Fries
6,330,593	B1	12/2001	Roberts et al.
6,345,256	B1	2/2002	Milsted et al.
6,349,296	B1	2/2002	Broder et al.
6,360,215	B1	3/2002	Judd et al.
6,363,377	B1	3/2002	Kravets et al.
6,374,225	B1	4/2002	Hejna, Jr.
6,374,260	B1	4/2002	Hoffert et al.
6,381,601	B1	4/2002	Fujiwara et al.
6,385,596	B1	5/2002	Wiser et al.
6,407,680	B1	6/2002	Lai et al.
6,408,128	B1	6/2002	Abecassis
6,415,280	B1	7/2002	Farber et al.
6,418,421	B1	7/2002	Hurtado et al.
6,438,556	B1	8/2002	Malik et al.
6,446,068	B1	9/2002	Kortge
6,449,226	B1	9/2002	Kumagai
6,452,874	B1	9/2002	Otsuka et al.
6,453,252	B1	9/2002	Laroche
6,460,050	B1	10/2002	Pace et al.
6,460,180	B1	10/2002	Park et al.
6,473,804	B1	10/2002	Kaiser et al.
6,477,704	B1	11/2002	Cremia
6,490,279	B1	12/2002	Chen et al.
6,496,802	B1	12/2002	Van Zoest et al.
6,496,857	B1	12/2002	Dustin et al.
6,505,160	B1	1/2003	Levy et al.
6,542,869	B1	4/2003	Foote
6,550,001	B1	4/2003	Corwin et al.
6,550,011	B1	4/2003	Sims, III
6,552,254	B2	4/2003	Hasegawa et al.
6,563,515	B1	5/2003	Reynolds et al.
6,564,379	B1	5/2003	Knudson et al.
6,571,392	B1	5/2003	Zigmond et al.
6,577,746	B1	6/2003	Evans et al.
6,591,245	B1	7/2003	Klug
6,598,228	B2	7/2003	Hejna, Jr.
6,604,242	B1	8/2003	Weinstein et al.
6,609,105	B2	8/2003	Van Zoest et al.
6,615,408	B1	9/2003	Kaiser et al.
6,631,523	B1	10/2003	Matthews, III et al.
6,654,757	B1	11/2003	Stern
6,665,661	B1	12/2003	Crow et al.
6,675,174	B1	1/2004	Bolle et al.
6,675,385	B1	1/2004	Wang
6,693,236	B1	2/2004	Gould et al.
6,698,020	B1	2/2004	Zigmond et al.
6,766,523	B2	7/2004	Herley
6,774,926	B1	8/2004	Ellis et al.
6,785,902	B1	8/2004	Zigmond et al.
6,810,388	B1	10/2004	Sato
6,833,865	B1	12/2004	Fuller et al.
6,834,308	B1	12/2004	Ikezoye et al.
6,871,200	B2	3/2005	MacQueen et al.
6,871,231	B2	3/2005	Morris
6,873,982	B1	3/2005	Bates et al.
6,928,423	B1	8/2005	Yamanaka
6,928,442	B2	8/2005	Farber et al.
6,931,451	B1	8/2005	Logan et al.
6,937,766	B1	8/2005	Wilf et al.
6,941,275	B1	9/2005	Swierczek
6,941,574	B1	9/2005	Broadwin et al.
6,944,632	B2	9/2005	Stern
6,968,337	B2	11/2005	Wold
6,978,419	B1	12/2005	Kantrowitz
6,978,461	B2	12/2005	Shapiro et al.
6,983,371	B1	1/2006	Hurtado et al.
6,990,453	B2	1/2006	Wang et al.
6,999,111	B2	2/2006	McIntyre et al.
7,013,301	B2	3/2006	Holm et al.
7,020,635	B2	3/2006	Hamilton et al.
7,039,935	B2	5/2006	Knudson et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,043,473 B1 5/2006 Rassool et al.
7,058,223 B2 6/2006 Cox
7,065,709 B2 6/2006 Ellis et al.
7,092,953 B1 8/2006 Haynes
7,096,486 B1 8/2006 Ukai et al.
7,103,906 B1 9/2006 Katz et al.
7,106,904 B2 9/2006 Shima
7,140,033 B1 11/2006 Durden et al.
7,146,631 B1 12/2006 Tanaka et al.
7,152,236 B1 12/2006 Wugofski et al.
7,155,449 B2 12/2006 Pingel et al.
7,158,929 B2 1/2007 Wouters et al.
7,165,266 B2 1/2007 Zigmond
7,168,083 B2 1/2007 Kalker et al.
7,171,016 B1 1/2007 Rhoads
7,174,293 B2 2/2007 Kenyon et al.
7,181,756 B1 2/2007 Zigmond et al.
7,184,100 B1 2/2007 Wilf et al.
7,188,353 B1 3/2007 Crinon
7,191,190 B2 3/2007 Debique et al.
7,225,455 B2 5/2007 Bennington et al.
7,237,253 B1 6/2007 Blackketter et al.
7,243,153 B2 7/2007 McIntyre et al.
7,254,829 B1 8/2007 Brown et al.
7,272,788 B2 9/2007 Anderson et al.
7,302,574 B2 11/2007 Conwell et al.
7,305,693 B2 12/2007 Blackketter et al.
7,308,413 B1 12/2007 Tota et al.
7,313,805 B1 12/2007 Rosin et al.
7,334,250 B2 2/2008 Blackketter et al.
7,340,763 B1 3/2008 Harris
7,346,472 B1 3/2008 Moskowitz et al.
7,363,278 B2 4/2008 Schmelzer et al.
7,366,718 B1 4/2008 Pugh et al.
7,366,787 B2 4/2008 Salas et al.
7,369,677 B2 5/2008 Petrovic et al.
7,370,017 B1 5/2008 Lindeman et al.
7,386,512 B1 6/2008 Allibhoy et al.
7,404,200 B1 7/2008 Hailey et al.
7,421,723 B2 9/2008 Harkness et al.
7,423,771 B2 9/2008 Ohata et al.
7,426,558 B1 9/2008 Allibhoy et al.
7,477,739 B2 1/2009 Haitsma et al.
7,483,958 B1 1/2009 Elabbady et al.
7,487,527 B2 2/2009 Ellis et al.
7,493,643 B2 2/2009 Ellis
7,500,007 B2 3/2009 Ikezoye et al.
7,506,352 B2 3/2009 Blackketter et al.
7,523,312 B2 4/2009 Kalker et al.
7,523,478 B2 4/2009 Blackketter et al.
7,529,659 B2 5/2009 Wold
7,562,012 B1 7/2009 Wold et al.
7,562,392 B1 7/2009 Rhoads et al.
7,565,327 B2 7/2009 Schmelzer
7,587,728 B2 9/2009 Wheeler et al.
7,595,914 B2 9/2009 Haining
7,606,883 B1 10/2009 Allibhoy et al.
7,624,337 B2 11/2009 Sull et al.
7,631,072 B2 12/2009 Allibhoy et al.
7,647,604 B2 1/2010 Ramaswamy
7,650,616 B2 1/2010 Lee
7,660,700 B2 2/2010 Moskowitz et al.
7,707,088 B2 4/2010 Schmelzer
7,711,652 B2 5/2010 Schmelzer
7,738,704 B2 6/2010 Lienhart et al.
7,757,248 B2 7/2010 Harkness et al.
7,757,254 B2 7/2010 Shoff et al.
7,765,575 B2 7/2010 Zigmond
7,783,489 B2 8/2010 Kenyon et al.
7,797,249 B2 9/2010 Schmelzer et al.
7,802,281 B1 9/2010 Tani et al.
7,818,768 B2 10/2010 Blackketter et al.
7,840,975 B2 11/2010 Matheny et al.
7,849,226 B2 12/2010 Zigmond et al.
7,853,664 B1 * 12/2010 Wang et al. 709/217
7,861,275 B1 12/2010 Vellaikal et al.
7,877,438 B2 1/2011 Schrempp et al.
7,882,518 B2 2/2011 Finseth et al.
7,917,645 B2 3/2011 Ikezoye
7,930,719 B2 4/2011 Ellis et al.
7,941,816 B2 5/2011 Harkness et al.
7,949,494 B2 5/2011 Moskowitz et al.
7,949,749 B2 5/2011 Allibhoy et al.
7,962,414 B1 6/2011 Allibhoy et al.
7,996,565 B2 8/2011 Allibhoy et al.
8,001,569 B2 8/2011 Marler et al.
8,006,264 B2 8/2011 Reynolds et al.
8,006,314 B2 8/2011 Wold
8,065,615 B2 11/2011 Murray et al.
8,082,150 B2 12/2011 Wold
8,086,445 B2 12/2011 Wold et al.
8,090,605 B2 1/2012 Tota et al.
8,094,949 B1 1/2012 Rhoads
8,108,886 B1 1/2012 Murahashi et al.
8,112,776 B2 2/2012 Schein et al.
8,171,509 B1 5/2012 Girouard et al.
8,171,510 B2 5/2012 Kamen et al.
8,185,923 B2 5/2012 Slaney et al.
8,214,175 B2 7/2012 Moskowitz et al.
RE43,578 E 8/2012 Sorensen
8,255,952 B2 8/2012 Boylan, III et al.
RE43,671 E 9/2012 Sorensen
8,272,011 B2 9/2012 Yuen et al.
8,296,792 B2 10/2012 Sahota et al.
8,301,758 B2 10/2012 Allibhoy et al.
8,340,994 B2 12/2012 Tota et al.
8,479,233 B2 7/2013 Ellis et al.
2001/0001160 A1 5/2001 Shoff et al.
2001/0003818 A1 6/2001 Pingel et al.
2001/0047298 A1 11/2001 Moore et al.
2001/0049625 A1 12/2001 Mowry
2002/0023020 A1 2/2002 Kenyon et al.
2002/0026369 A1 2/2002 Miller et al.
2002/0032698 A1 3/2002 Cox
2002/0038296 A1 3/2002 Margolus et al.
2002/0056123 A1 5/2002 Liwerant et al.
2002/0059610 A1 5/2002 Ellis
2002/0082731 A1 6/2002 Pitman et al.
2002/0083005 A1 6/2002 Lowenstein et al.
2002/0087885 A1 7/2002 Peled et al.
2002/0120925 A1 8/2002 Logan
2002/0133499 A1 9/2002 Ward et al.
2002/0150164 A1 10/2002 Felts et al.
2002/0156760 A1 10/2002 Lawrence et al.
2002/0178276 A1 11/2002 McCartney et al.
2002/0186887 A1 12/2002 Rhoads
2003/0028489 A1 2/2003 Williamson
2003/0037010 A1 2/2003 Schmelzer
2003/0061490 A1 3/2003 Abajian
2003/0093790 A1 5/2003 Logan et al.
2003/0095660 A1 5/2003 Lee et al.
2003/0101144 A1 5/2003 Moreno
2003/0106017 A1 6/2003 Kanchirayappa et al.
2003/0146940 A1 8/2003 Ellis et al.
2003/0202660 A1 10/2003 Zhou et al.
2003/0233930 A1 12/2003 Ozick
2004/0003398 A1 1/2004 Donian et al.
2004/0010602 A1 1/2004 Van Vleck et al.
2004/0015608 A1 1/2004 Ellis et al.
2004/0025174 A1 2/2004 Cerrato et al.
2004/0163106 A1 8/2004 Schrempp et al.
2004/0170335 A1 9/2004 Pearlman et al.
2004/0199387 A1 * 10/2004 Wang et al. 704/243
2004/0221118 A1 11/2004 Slater et al.
2004/0243540 A1 12/2004 Moskowitz et al.
2005/0044189 A1 2/2005 Ikezoye et al.
2005/0080846 A1 4/2005 McCleskey et al.
2005/0102515 A1 5/2005 Jaworski et al.
2005/0154892 A1 7/2005 Mihcak et al.
2005/0160363 A1 7/2005 Bhogal et al.
2005/0193016 A1 9/2005 Seet et al.
2005/0213826 A1 9/2005 Neogi
2005/0246752 A1 11/2005 Liwerant et al.
2005/0289065 A1 12/2005 Weare

(56) **References Cited**

U.S. PATENT DOCUMENTS

2006/0031870	A1	2/2006	Jarman et al.
2006/0080356	A1	4/2006	Burges et al.
2006/0085816	A1	4/2006	Funk et al.
2006/0101069	A1	5/2006	Bell et al.
2006/0110137	A1	5/2006	Tsuda et al.
2006/0187358	A1	8/2006	Lienhart et al.
2006/0195859	A1	8/2006	Konig et al.
2006/0195860	A1	8/2006	Eldering et al.
2006/0206462	A1	9/2006	Barber
2006/0212927	A1	9/2006	Riku et al.
2006/0271947	A1	11/2006	Lienhart et al.
2007/0041667	A1	2/2007	Cox
2007/0071330	A1	3/2007	Oostveen et al.
2007/0083510	A1	4/2007	McArdle
2007/0101360	A1	5/2007	Gutta et al.
2007/0118375	A1	5/2007	Kenyon et al.
2007/0124698	A1	5/2007	Majumder
2007/0130580	A1	6/2007	Covell et al.
2007/0180537	A1	8/2007	He et al.
2007/0203911	A1	8/2007	Chiu
2007/0282472	A1	12/2007	Seldman
2007/0288518	A1	12/2007	Crigler et al.
2008/0052783	A1	2/2008	Levy
2008/0091684	A1	4/2008	Ellis et al.
2008/0250241	A1	10/2008	Ginter et al.
2009/0052784	A1	2/2009	Covell et al.
2010/0211969	A1	8/2010	Schein et al.
2011/0167449	A1	7/2011	Klosterman et al.
2011/0173660	A1	7/2011	Schein et al.
2013/0086608	A1	4/2013	Slaney et al.

FOREIGN PATENT DOCUMENTS

EP	1 485 815	B1	7/2009
GB	2369203	A	5/2002
JP	2003-242281		8/2003
WO	94/06084	A1	3/1994
WO	99/50778	A1	10/1999
WO	WO0122730	A1	3/2001
WO	WO 02/011033	A1	2/2002
WO	WO 02/103968	A1	12/2002

OTHER PUBLICATIONS

Martin Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996.
Yossi Rubner et al., "Adaptive Color Image Embeddings for Database Navigation*," Proceedings of the 1998 IEEE Asian Conference on Computer Vision.
Roger Weber et al., "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," Proceedings of 24th VLDB Conference, 1998.
P. Yianilos, "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces," Proceedings of the ACM-SIAM Symposium on Discrete algorithms, 1993, pp. 311.321.
U.S. Appl. No. 60/304,647, filed Jul. 10, 2001.
U.S. Appl. No. 60/281,881, filed Apr. 5, 2001.
Metadata Mediation: Representation and Protocol, Tsuyoshi Sakata, Hiroyuki Tada, Tomohisa Ohtake, Digital Vision Laboratories, Jul. 3, 1937 Akasaka, Minato, Tokyo, Japan.
U.S. Appl. No. 13/800,573, filed Mar. 13, 2013.
U.S. Appl. No. 13/800,890, filed Mar. 13, 2013.
U.S. Appl. No. 13/829,717, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,447, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,626, filed Mar. 14, 2013.
U.S. Appl. No. 13/830,986, filed Mar. 14, 2013.
U.S. Appl. No. 13/842,068, filed Mar. 15, 2013.
U.S. Appl. No. 13/338,079, filed Dec. 27, 2011.
U.S. Appl. No. 60/134,782, filed May 19, 1999.
U.S. Appl. No. 60/193,948, filed Mar. 31, 2000.
U.S. Appl. No. 60/195,535, filed Apr. 7, 2000.
U.S. Appl. No. 60/206,384, filed May 23, 2000.
U.S. Appl. No. 60/221,843, filed Jul. 28, 2000.
Peter N. Yianlos, Excluded Middle Vantage Point Forests for Nearest Neighbor Search, Jul. 20, 1998, pp. 1-12.
Peter N. Yianlos,, "Locally Lifting the Curse of Dimensionality for Nearest Neighbor Search" SODA 2000, pp. 361-370.
L. Baum et al., "A Maximation Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains," The Annals of Mathematical Statistics, vol. 41, No. 1, pp. 164-171 (1970).
A. P. Dempster et al., "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, Series B (Methodological), vol. 39, Issue 1, pp. 1-38 (1977).
D. Reynolds et al., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," IEEE Transactions on Speech and Audio Processing, vol. 3, No. 1, pp. 72-83 (Jan. 1995).
D. Bouktache, "A fast Algorithm for the nearest neighbor classifier," IEEE Transactions on Pattern Analysis and Machine Intelligence, Mar. 1997, pp. 277-282.
Nene et al., "A simple algorithm for nearest neighbor search in high dimensions," IEEE Transactions on Pattern Analysis and Machine Intelligence, Sep. 1997, pp. 989-1003.
Sunil Arya et al., "Approximate Nearest Neighbor Queries in Fixed Dimensions," Proceedings of the 4th annual ACM-SIAM Symposium on Discrete algorithms, 1993, pp. 271-280.
K. Fukunaga et al., A branch and bound algorithm for computing k-nearest neighbors, IEEE Trans. Comput., C24:750-753, Jul. 1975.
Charles D. Feustel et al., "The nearest neighbor problem in an abstract metric space," Pattern Recognition Letters, pp. 125-128, Dec. 1982.
Dennis Shasha et al., "New Techniques for Best-Match Retrieval," ACM Transactions on Information Systems, 8(2); 140{158, Apr. 1990.
J. Uhlmann, "Satisfying general proximity / similarity queries with metric trees", Information Processing Letters, 40 (4): 175{9, Nov. 1991.
Sergey Brin, "Near Neighbor Search in Large Metric Spaces," Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995.
Daniel P. Huttenlocher, et al., "Comparing Images Using the Hausdorff Distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, No. 9, pp. 850-863, Sep. 1993.
Thomas Seidl et al., "Optimal Multi-Step K-Nearest Neighbor Search," Proceedings of ACM SIGMOD International Conference of Management of Data, Jun. 1998, pp. 154-165.
W. A. Burkhard et al., "Some Approaches to Best-Match File Searching," Communications of the ACM, vol. 16, No. 4, Apr. 1973.
Eyal Kushilevitz et al., "Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces," Proceedings of the 30th annual ACM Symposium on the Theory of computing, 1998, pp. 457-474, vol. 30, No. 2.
J. Nievergelt et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," ACM Transactions on Database Systems, vol. 9, No. 1, pp. 38-71 (Mar. 1984).
Nevin Heintze, "Scalable Document Fingerprinting," Proc. USENIX Workshop on Electronic Commerce (1996).
Erling Wold et al., "Content-Based Classification, Search, and Retrieval of Audio," IEEE Multimedia, vol. 3, Issue 3, pp. 27-63 (1996).
Bir Bhanu et al., "Learning Feature Relevance and Similarity Metrics in Image Databases," Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 14-19 (1998).
A. Del Bimbo et al., "Using Weighted Spatial Relationships in Retrieval by Visual Contents," Image Description and Retrieval, pp. 161-192 (1998).
P. Indyk et al "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," Proceeding of The Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604-613 Jul. 21, 1999.
Marco La Cascia, "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web," Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 24-29 (1998).

(56) **References Cited**

OTHER PUBLICATIONS

Atsuo Yoshitaka et al., "A Survey on Content-Based Retrieval for Multimedia Databases," IEEE Transactions on Knowledge and Data Engineering, vol. 11, No. 1, pp. 81-93 (Jan./Feb. 1999).
Steve Lawrence et al., "Digital Libraries and Automonous Citation Indexing," IEEE Computer, pp. 67-71 (Jun. 1999).
Akisato Kimura et al., "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," IEEE Conf on Acoustics, Speech and Signal Processing, (ICASSP '01), vol. 3, pp. 1429-1432, 2001.
Edgar Chavez et al., "Searching in Metric Spaces, ACM Computing Surveys," vol. 33, No. 3, pp. 273-321 (Sep. 2001).
Jaap Haitsma et al., "Robust Audio Hashing for Content Identification," Workshop on Content Based Multimedia Indexing, Brescia, Italy (Sep. 19-21, 2001).
Jaap Haitsma et al., "A Highly Robust Audio Fingerprinting System." Journal of New Music Research, 1744-5027, vol. 32, Issue 2, pp. 211-221 (2003).
Saul Schleimer et al., "Winnowing: Local Algorithms for Document Fingerprinting," ACM SIGMOD, Jun. 9-12, 2003.
Edward Chang et al., "Searching Near-Replicas of Images via Clustering," SPIE Symposium of Voice, Video and Data Communications, 1999.
Edward Y. Chang et al., "RIME: A Replicated Image Detector for the World-Wide Web," SPIE, 1998.
Hector Garcia-Molina et al., "Safeguarding and Charging for Information on the Internet," Proceedings of ICDE, 1998.
Sergey Brin et al., "Copy Detection Mechanisms for Digital Documents," Proceedings of ACM SIG-MOD, May 1995.
Stefan Berchtold "The x-tree: An Index Structure for High-Dimensional Data," Proceedings of the 22nd VLDB, Aug. 1996.
Norio Katayama et al., "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proceedings of ACM SIGMOD, May 1997.
John T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," Proceedings of ACM SIGMOD, Apr. 1981.
Myron Flickner et al., "Query by Image and Video Content: The QBIC System," IEEE Computer 28(9), pp. 23-32, 1995.
Amarnath Gupta et al., "Visual Information Retrieval," Communications of the ACM, vol. 40, No. 5, pp. 69-79, May 1997.
John R. Smith et al., "VisualSEEk: A fully automated content-based image query system," ACM Multimedia Conference, 1996.
David A. White et al., "Similarity Indexing: Algorithms and Performance," Proc. SPIE, vol. 2670, San Diego, 1996.
Norbert Beckmann et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," Proceedings of ACM Sigmod, May 1990.
A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proceedings of ACM Sigmod, Jun. 1984.
David A. White et al., "Similarity Indexing with the SS-tree*," Proceedings of the 12th ICDE, Feb. 1996.
King Lin et al., "The TV-Tree: An Index Structure for High-Dimensional Data," VLDB, Journal 3, No. 4, 1994, pp. 517-542.
Paolo Ciaccia et al., "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," Proceedings of the 23rd VLDB, Aug. 1997.
Nick Roussopoulos et al., Nearest Neighbor Queries, Proceedings of ACM Sigmod, May 1995.
C. Li et al., "An extensible hashing index for high-dimensional similarity search," Stanford Techncial Report, Aug. 1998.
Jon M. Kleinberg, "Two Algorithms for Nearest-Neighbor Search in High Dimensions," Proc 29th STOC, Feb. 7, 1997.
U.S. Appl. No. 60/222,023, filed Jul. 31, 2000; Avery Li-Chun Wang and Julius O. Smith III, Inventors; Palo Alto, CA.
Liu et al., "An Investigation of Practical Approximate Nearest Neighbor Algorithms", Advances in Neural Information Processing Systems (NIPS) 2004.
K.L. Clarkson, "Nearest-Neighbor Searching and Metric Space Dimensions", Nearest-Neighbor Methods for Learning and Vision: Theory and Practice:, Apr. 2005.
Swaminathan et al., "Robust an Secure Image Hashing", IEEE Transactions on Information Forensics and Security, Jun. 2006, pp. 215-230, vol. 1, No. 2.
Burges et al., "Duplicate Detection and Audio Thumbnails with Audio Fingerprinting" [online]. 2004, [retrieved on Nov. 21, 2006]. Retrieved on the Internet: <URL:www.research.microsoft.com/~cburges/tech_reports/tr-2004-19.pdf>, 5 pages.
Cano et al., "A Review of Algorithms for Audio Fingerprinting" [online]. 2002, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.iua.upf.es/mtg/publications/MMSP-2002-pcano.pdf>, 5 pages.
Haitsma and Kalker, "A Highly Robust Audio Fingerprinting System" [online]. 2002, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <url:www.ismir2002.ismir.net/proceedings/02-FP04-2.pdf>, 9 pages.
Jacobs et al., "Fast Multiresolution Image Querying" [online]. 1995, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.grail.cs.washington.edu/projects/query.pdf>, 10 pages.
Ke et al., "Computer Vision for Music Identification" [online]. 2005, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.cs.cmu.edu/~yke/musicretrieval/cvpr2005-mr.pdf>, 8 pages.
Shazam, Shazam Entertainment Brings Music Recognition to Windows Mobile 5.0 Powered Smartphones [online]. 2006. [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <URL: www.shazam.com/music/portals/sp/s/media-type/html/user/anon/page/default/template/pages/p/company_release30.html>, 1 page.
Stanford, "CS276 Information Retrieval and Web Mining" [online]. 2005, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <URL: www.stanford.edu/class/cs276/handouts/lecture19.pdf>, 8 pages.
Stanford, "Data Mining: Associations" [online]. 2002, [retrieved on Nov. 16, 2006]. Retrieved from the Internet: <URL: www.stanford.edu/class/cs206/cs206-2.pdf>, 11 pages.
Stollnitz et al., Wavelets for Computer Graphics: A Primerr, Part 1: [online]. 1995, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL: www.grail.cs.washington.edu/pub/stoll/wavelet1.pdf>, 8 pages.
Stollnitz et al., Wavelets for Computer Graphics: A Primer, Part 2: [online]. 1995, [retrieved on Nov. 21, 2006]. Retrieved from the Internet: <URL:www.grail.cs.washington.edu/pub/stoll/wavelet2.pdf>, 9 pages.
Yang, "MACS: Music Audio Characteristic Sequence Indexing for Similarity Retrieval", Oct. 21-24, 2001, New Paltz, New York.
Viola and Jones, Robust Real-Time Object Detection, Int. J. Computer Vision, 2002.
Burges et al., "Using Audio Fingerprinting for Duplicate Detection and Thumbnail Generation," Mar. 2005, 4 pages.
Lin, et al., Input Data Representation for Self-Organizing Map in Software Classification, Knowledge Acquisition and Modeling, 2009. KAM '09. Second International Symposium on Volume: 2, Digital Object Identified: 10.1109/KAM.2009.151 Publication Year: 2009, pp. 360-353.
Baluja et al., "Content Fingerprinting Using Wavelets", 3rd European Conference on Visual Media Production, 2006, pp. 198-207.
Cohen et al., "Finding Interesting Associations without Support Pruning", IEEE Transactions on Knowledge and Data Engineering, 2001, pp. 64-78, vol. 13, Issue 1.
Yang, Efficient Video Identification based on locality sensitive hashing and triangle inequality, National University of Singapore, 2005, pp. 1-64.
Ardizzone, Edoardo et al., "Motion and Color-Based Video Indexing and Retrieval," Universita di palermo, Departimento di Ingegneria Elettrica, pp. 135-139, Viale delle Scienze, Palermo, Italy, IEEE 1996.
Deng, Yining et al., "Content-based Search of Video Using Color, Texture, and Motion," Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA, pp. 534-537, IEEE 1997.
Fang, Min et al., "Computing Iceberg Queries Efficiently," Dept. of Computer Science, Stanford, CA, Paper No. 234, pp. 1-25.

(56) **References Cited**

OTHER PUBLICATIONS

Flickner, Myron et al., "Query by Image and Video Content: The QBIC System," IBM Almaden Research Center, Sep. 1995, pp. 23-32, IEEE 1995.
Gargi, U et al., "Performance Characterization and Comparison of Video Indexing Algorithms," Dept. of Computer Science & Engineering, the Pennsylvania State University, University Park, PA.
Gionis, Aristides et al., "Similarity Search in High Dimensions via Hashing," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 518-529, Proceeding of the 25th VLDB Conference, Edinburgh, Scotland, 1999.
Indyk, Piotr et al., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality" (preliminary version) Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13 & i-vii, Jul. 21, 1999.
Iyengar, Giridharan et al., "Models for automatic classification of video sequences," MIT Media Laboratory, Cambridge, MA.
Jain, Anil K., et al., "Image Retrieval using Color and Shape," Dept. of Computer Science, Michigan State University, Eas Lansing, MI, pp. 1-24, May 15, 1995.
Ogle, Virginia E., et al., "Chabot: Retrieval from a Relational Database of Images," University of California at Berkeley, Computer pp. 40-48, IEEE 1995.
Pentland, A. et al., "Photobook: Content-Based Manipulation of Image Databases," Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Tech., International Journal of Computer Vision 18(3), pp. 233-254 (1996), 1996 Kluwer Academic Publishers. Manuf. in The Netherlands.
Shivakumar, Narayanan et al., "SCAM: A Copy Detection Mechanism for Digital Documents," Dept. of Computer Science, Stanford University, Stanford, CA, pp. 1-13.
Shivakumar, Narayanan et al., "Building a Scalable and Accurate Copy Detection Mechanism," Dept. of Computer Science, Stanford University, Stanford, CA.
Srihari, Rohini K., "Automatic Indexing and Content-Based Retrieval of Captioned Images," State University of New York, Buffalo, Theme Feature, pp. 49-56, Sep. 1995, IEEE 1995.
Swain, Michael and Ballard, Dana H., "Color Indexing," International Journal of Computer Vision 7:1, p. 11-32 (1991), 1991 Kluwer Academic Publishers. Manuf. in The Netherlands.
Wactlar, Howard D. et al., "Intelligent Access to Digital Video: Informedia Project," Carnegie Mellon University, Digital Library Initiative: Carnegie Mellon University, Computer, pp. 46-52, IEEE 1996.
Yeo, Boon-Lock et al., "Rapid Scene Analysis on Compressed Video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, No. 6, pp. 533-544, Dec. 1995, Dept. of Electrical Engineering, Princeton University, Princeton, NJ, IEEE Log No. 9415901.
Indyk, Piotr et al., "Finding pirated video sequences on the Internet," Dept. of Computer Science, Stanford University, Palo Alto, CA, Paper No. 199.
U.S. Appl. No. 60/133,247, filed May 5, 1999.
U.S. Appl. No. 60/155,064, filed Sep. 21, 1999.
U.S. Appl. No. 60/218,824, filed Jul. 18, 2000.
Indyk, Piotr et al., "Locality-Preserving Hashing in Multidimensional Spaces," Feb. 25, 1997.
Gibson, David, "Name That Clip: Music retrieval using audio clips," Aug. 19, 1999.
Declaration of David A. Gibson, Inter Partes Review of U.S. Patent 7,174,293, Aug. 30, 2013.
Declaration of David A. Gibson, Inter Partes Review of U.S. Patent 7,783,489, Aug. 30, 2013.
Intersil, "Glossary of Communication Terms," Dec. 1996.
Declaration of Dr. Ton Kalker, Inter Partes Review of U.S. Patent 7,174,293, Aug. 30, 2013.
Declaration of Dr. Ton Kalker, Inter Partes Review of U.S. Patent 7,783,489, Aug. 30, 2013.

* cited by examiner

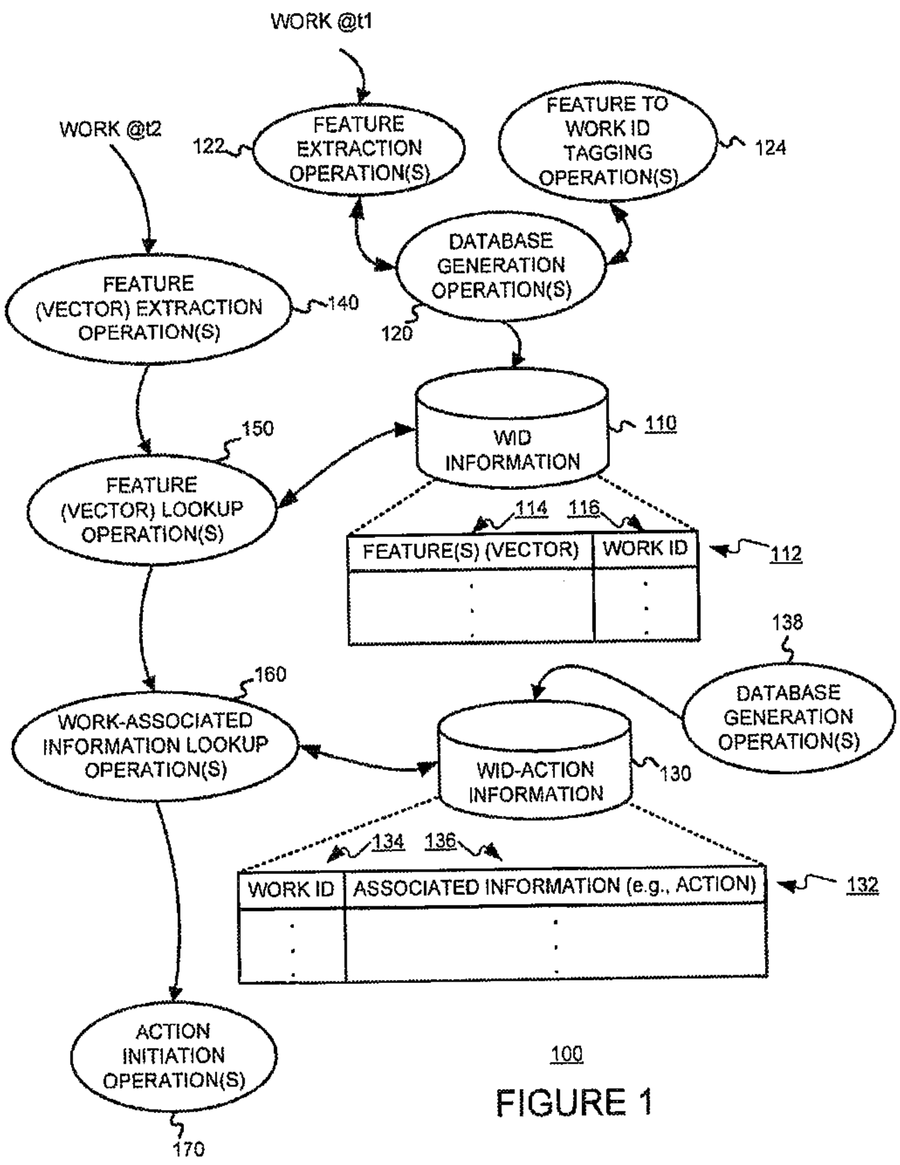

WORK @t1
WORK @t2
122
FEATURE EXTRACTION OPERATION(S)
FEATURE TO WORK ID TAGGING OPERATION(S)
124
DATABASE GENERATION OPERATION(S)
120
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140
150
FEATURE (VECTOR) LOOKUP OPERATION(S)
WID INFORMATION
110
114
116
FEATURE(S) (VECTOR)
WORK ID
112
138
DATABASE GENERATION OPERATION(S)
160
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
WID-ACTION INFORMATION
130
134
136
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
132
ACTION INITIATION OPERATION(S)
170
100

FIGURE 1

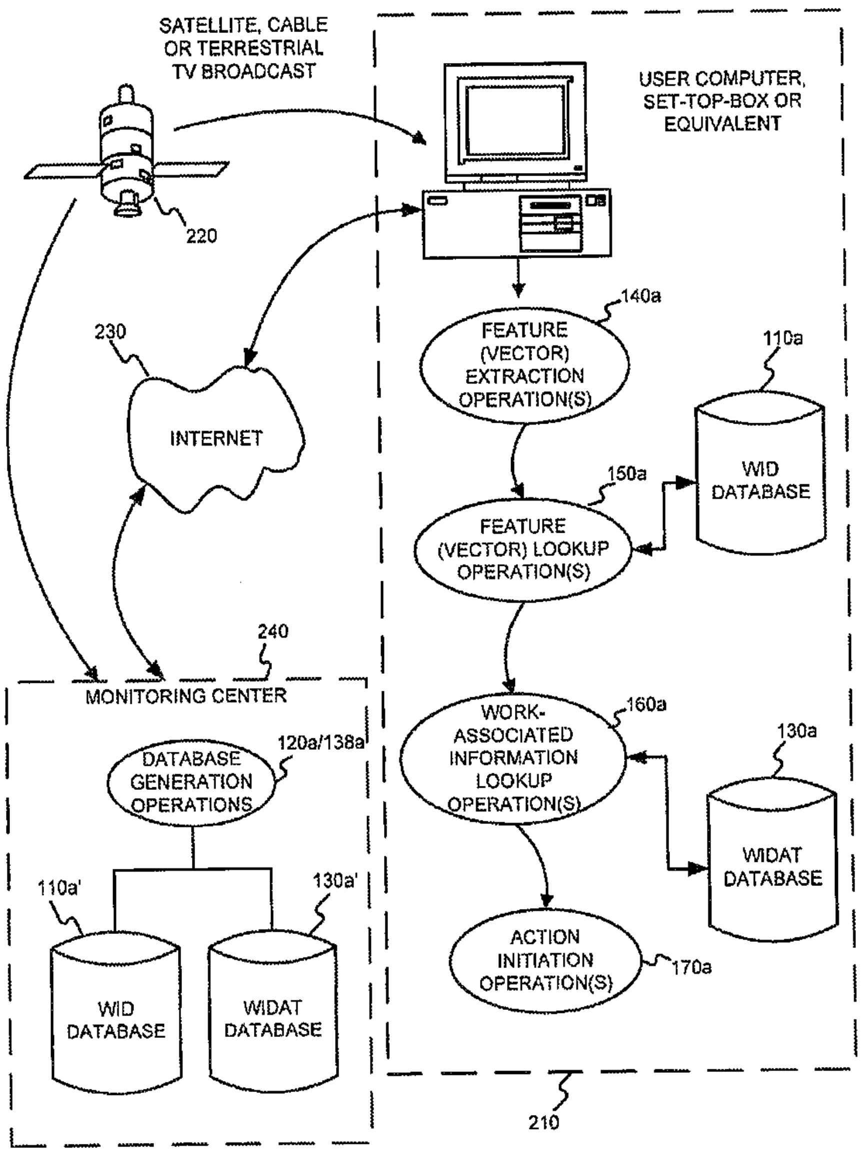

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
220
140a
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
230
110a
INTERNET
150a
WID
DATABASE
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
240
MONITORING CENTER
WORK-
ASSOCIATED
INFORMATION
LOOKUP
OPERATION(S)
160a
130a
DATABASE
GENERATION
OPERATIONS
120a/138a
110a'
130a'
WIDAT
DATABASE
ACTION
INITIATION
OPERATION(S)
170a
WID
DATABASE
WIDAT
DATABASE
210

FIGURE 2

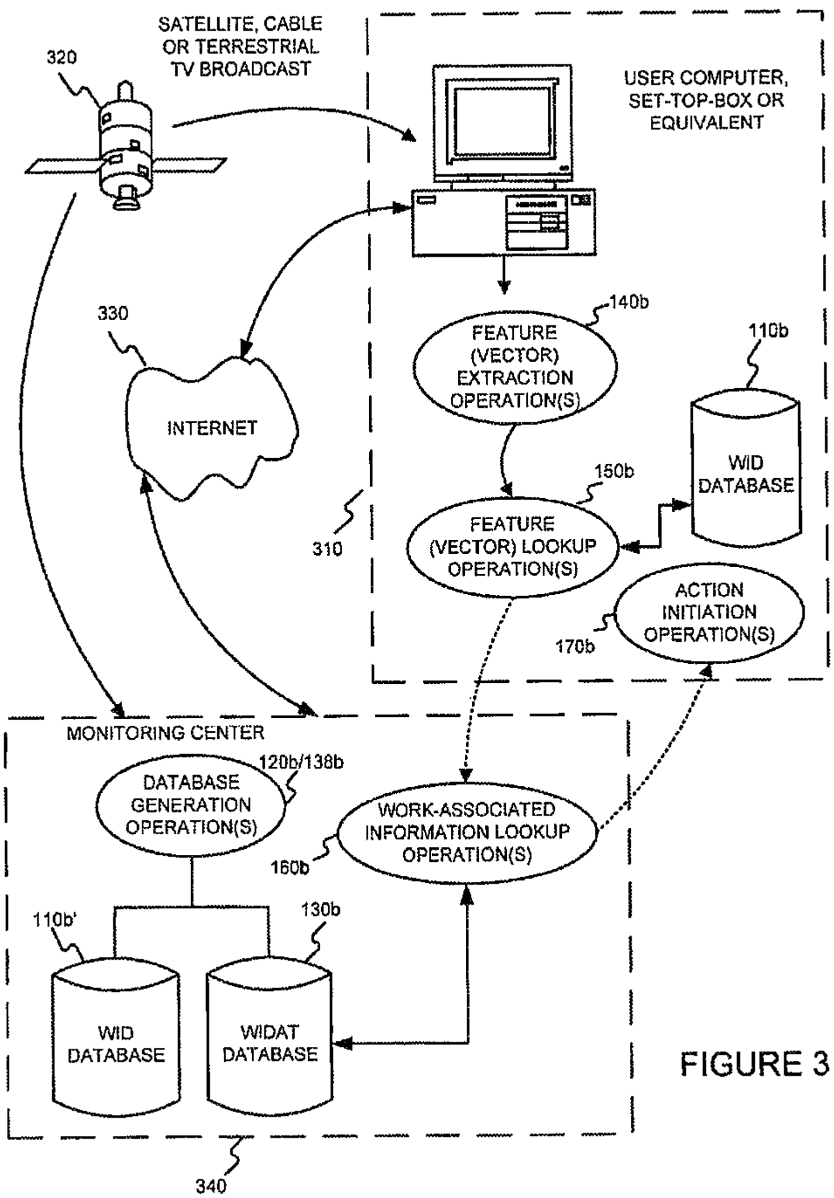

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
320
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
330
INTERNET
140b
FEATURE (VECTOR) EXTRACTION OPERATION(S)
110b
WID DATABASE
150b
FEATURE (VECTOR) LOOKUP OPERATION(S)
310
ACTION INITIATION OPERATION(S)
170b
MONITORING CENTER
120b/138b
DATABASE GENERATION OPERATION(S)
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
160b
110b'
130b
WID DATABASE
WIDAT DATABASE
340

FIGURE 3

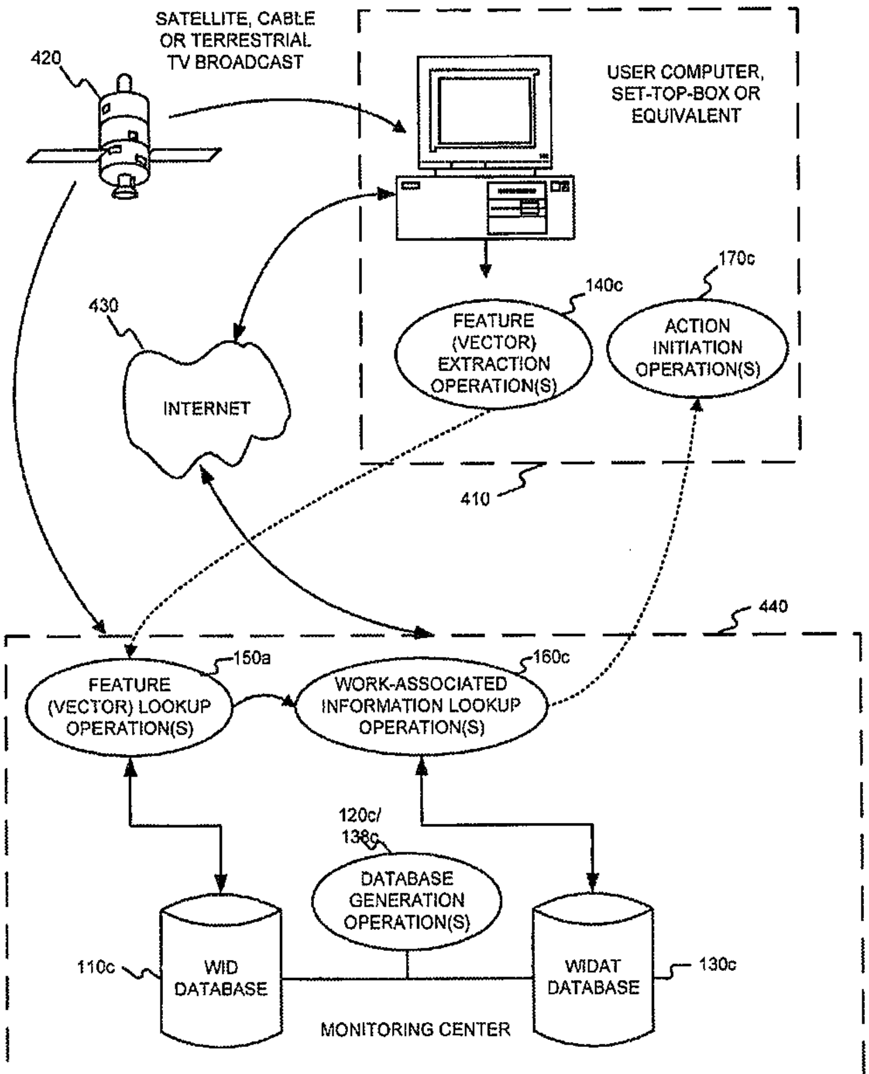

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
420
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
170c
140c
430
FEATURE
(VECTOR)
EXTRACTION
OPERATION(S)
ACTION
INITIATION
OPERATION(S)
INTERNET
410
440
150a
160c
FEATURE
(VECTOR) LOOKUP
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
120c/
138c
DATABASE
GENERATION
OPERATION(S)
110c
WID
DATABASE
WIDAT
DATABASE
130c
MONITORING CENTER

FIGURE 4

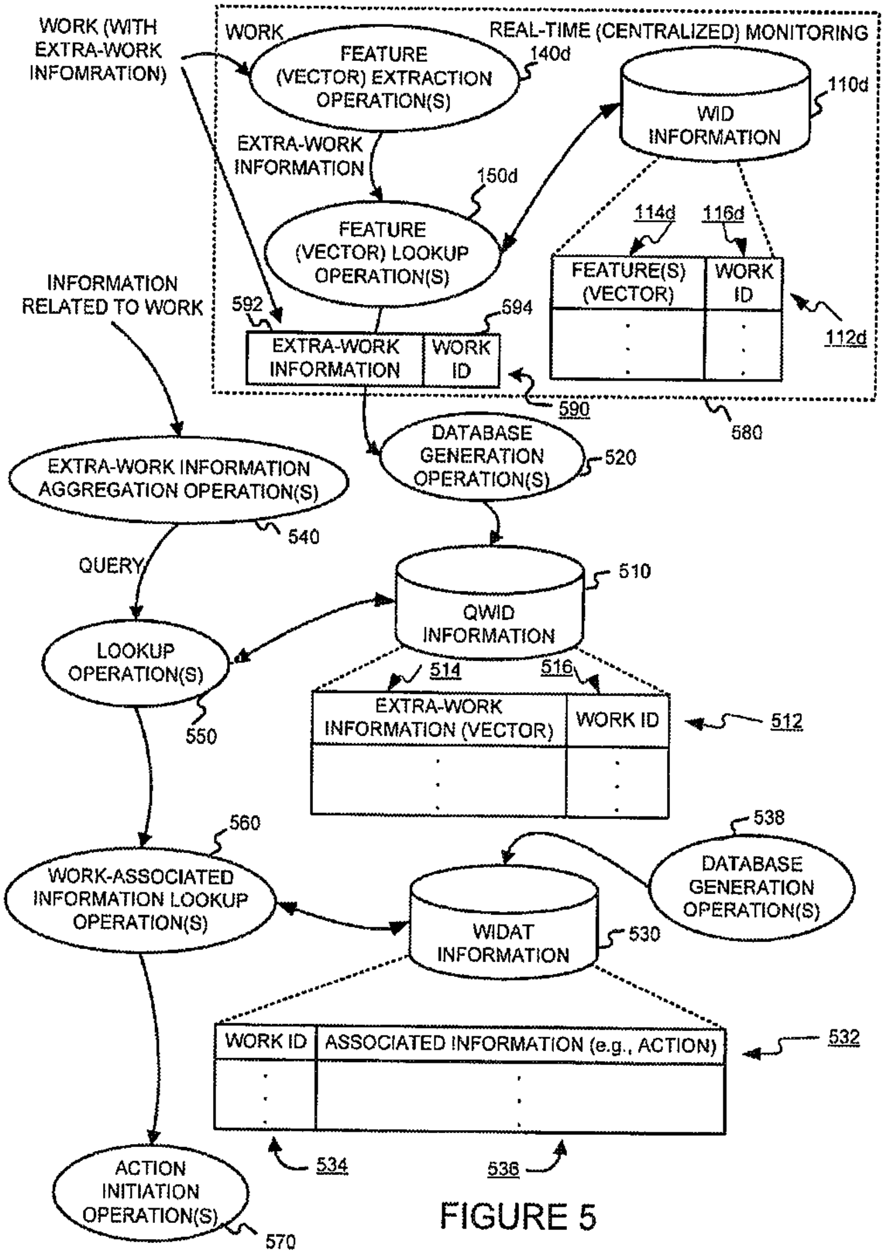

WORK (WITH EXTRA-WORK INFOMRATION)
WORK
REAL-TIME (CENTRALIZED) MONITORING
FEATURE (VECTOR) EXTRACTION OPERATION(S)
140d
WID INFORMATION
110d
EXTRA-WORK INFORMATION
150d
FEATURE (VECTOR) LOOKUP OPERATION(S)
114d
116d
FEATURE(S) (VECTOR)
WORK ID
112d
INFORMATION RELATED TO WORK
592
594
EXTRA-WORK INFORMATION
WORK ID
590
580
DATABASE GENERATION OPERATION(S)
520
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
540
QUERY
510
QWID INFORMATION
LOOKUP OPERATION(S)
514
516
EXTRA-WORK INFORMATION (VECTOR)
WORK ID
512
550
560
538
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
DATABASE GENERATION OPERATION(S)
WIDAT INFORMATION
530
WORK ID
ASSOCIATED INFORMATION (e.g., ACTION)
532
534
536
ACTION INITIATION OPERATION(S)
570

FIGURE 5

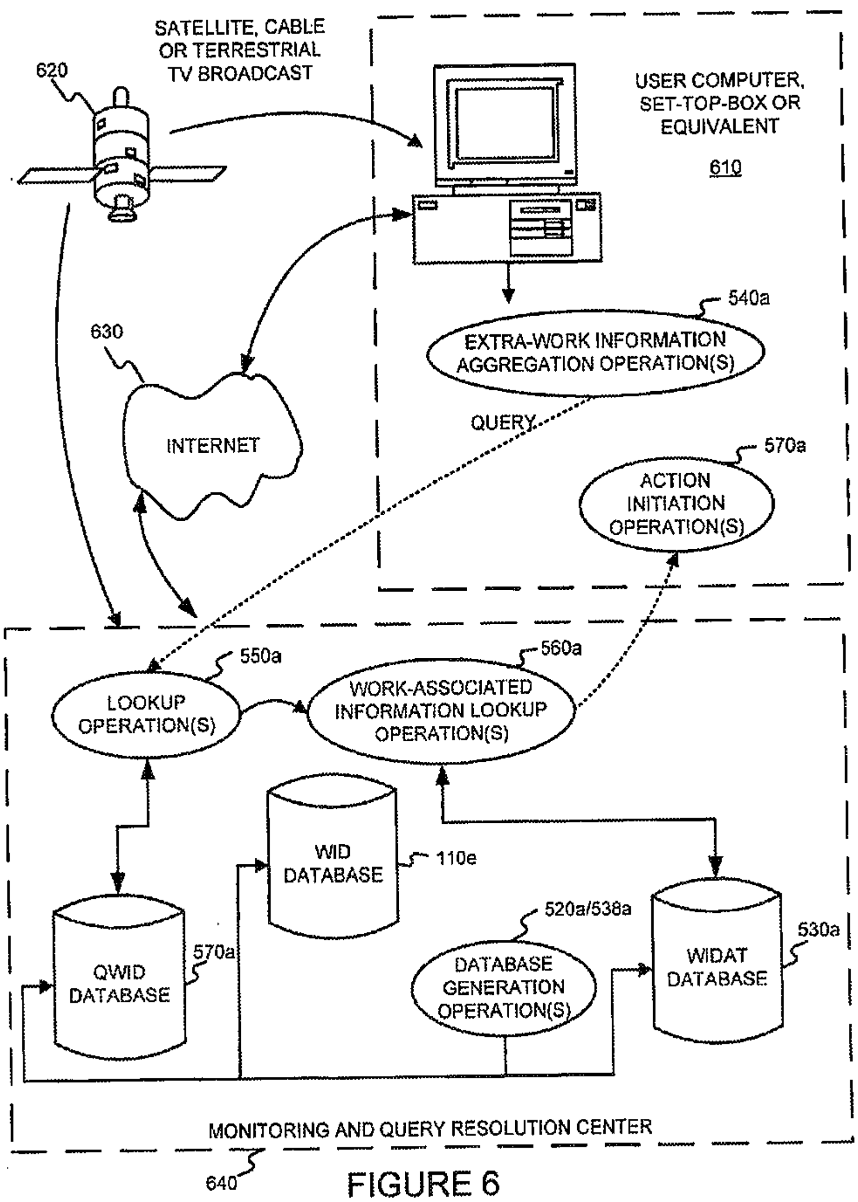

SATELLITE, CABLE
OR TERRESTRIAL
TV BROADCAST
620
USER COMPUTER,
SET-TOP-BOX OR
EQUIVALENT
610
540a
EXTRA-WORK INFORMATION
AGGREGATION OPERATION(S)
630
INTERNET
QUERY
570a
ACTION
INITIATION
OPERATION(S)
550a
560a
LOOKUP
OPERATION(S)
WORK-ASSOCIATED
INFORMATION LOOKUP
OPERATION(S)
WID
DATABASE
110e
520a/538a
530a
QWID
DATABASE
570a
DATABASE
GENERATION
OPERATION(S)
WIDAT
DATABASE
MONITORING AND QUERY RESOLUTION CENTER
640

FIGURE 6

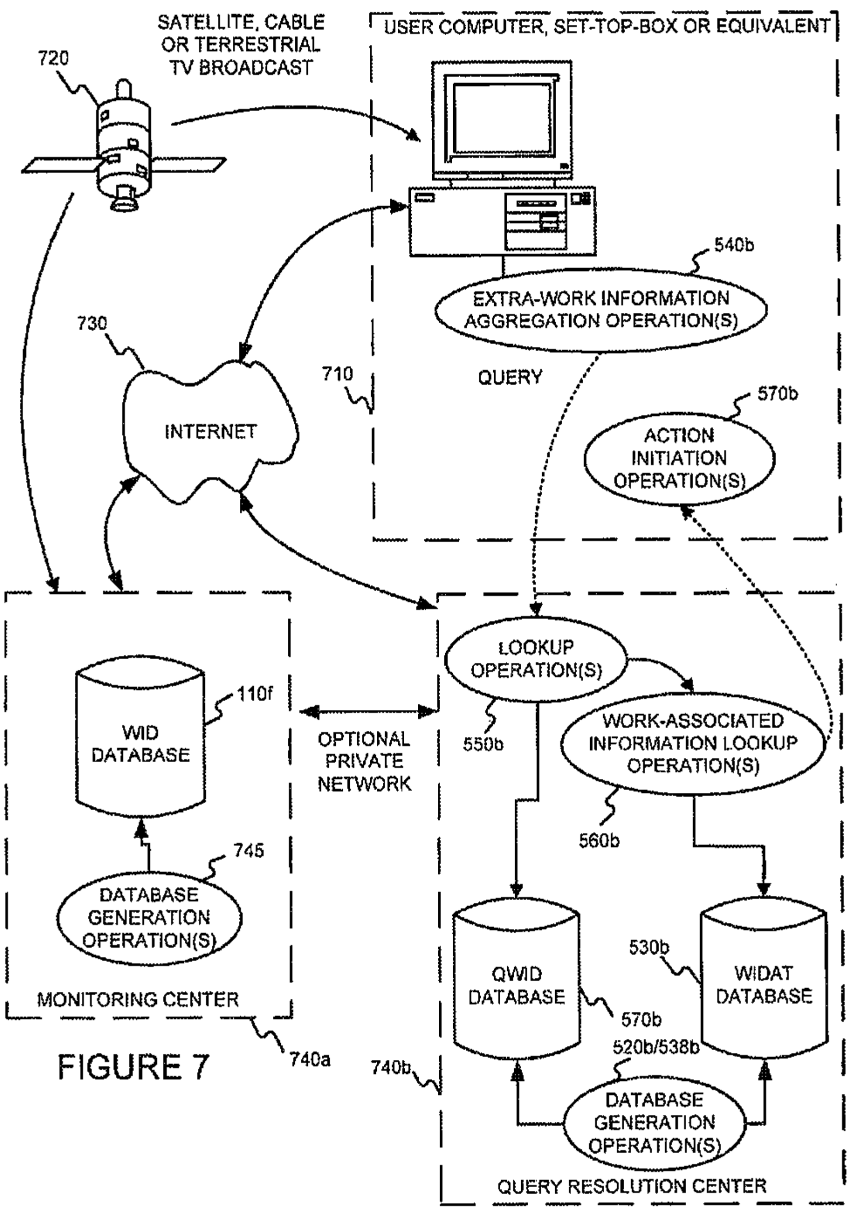
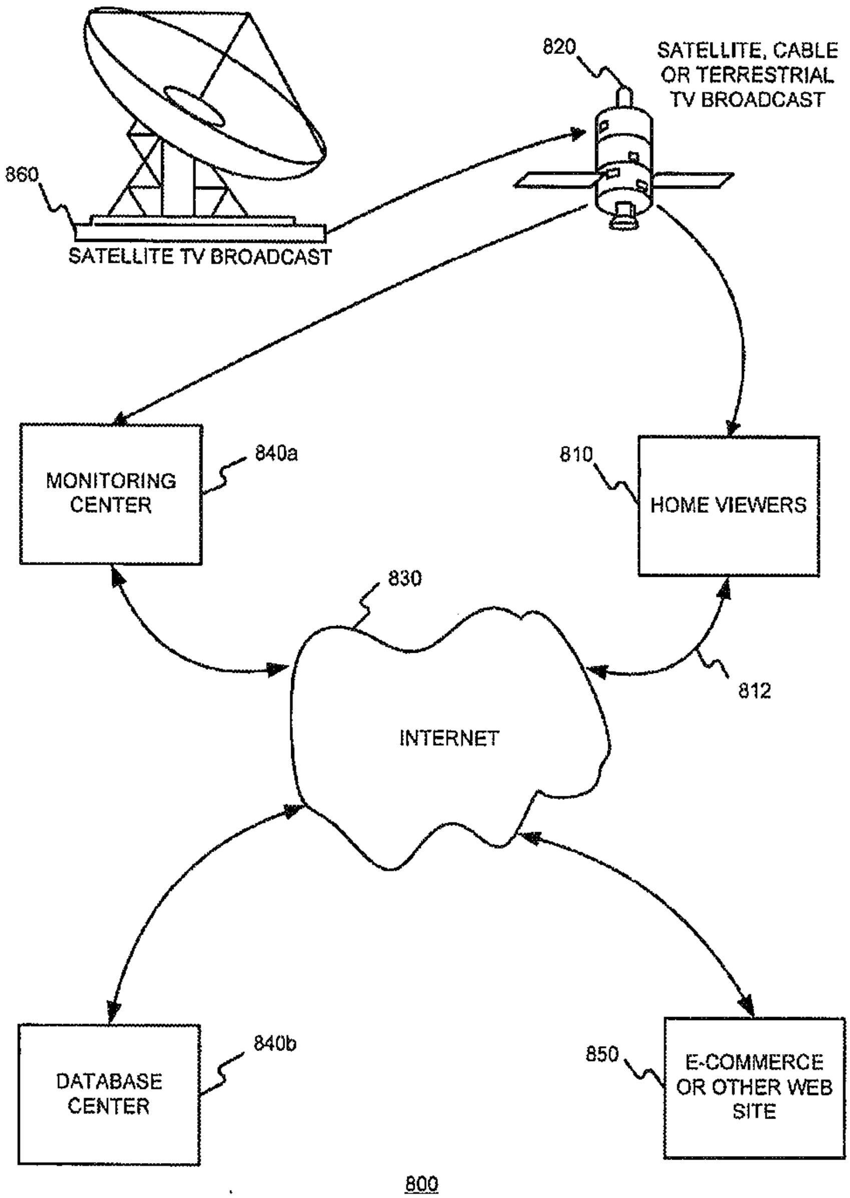

SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
720
USER COMPUTER, SET-TOP-BOX OR EQUIVALENT
540b
EXTRA-WORK INFORMATION AGGREGATION OPERATION(S)
730
710
QUERY
570b
INTERNET
ACTION INITIATION OPERATION(S)
LOOKUP OPERATION(S)
110f
WID DATABASE
OPTIONAL PRIVATE NETWORK
550b
WORK-ASSOCIATED INFORMATION LOOKUP OPERATION(S)
560b
745
DATABASE GENERATION OPERATION(S)
530b
QWID DATABASE
WIDAT DATABASE
MONITORING CENTER
570b
520b/538b
740a
740b
DATABASE GENERATION OPERATION(S)
QUERY RESOLUTION CENTER

FIGURE 7



860
SATELLITE TV BROADCAST
820
SATELLITE, CABLE OR TERRESTRIAL TV BROADCAST
MONITORING CENTER
840a
810
HOME VIEWERS
830
812
INTERNET
DATABASE CENTER
840b
850
E-COMMERCE OR OTHER WEB SITE
800

FIGURE 8

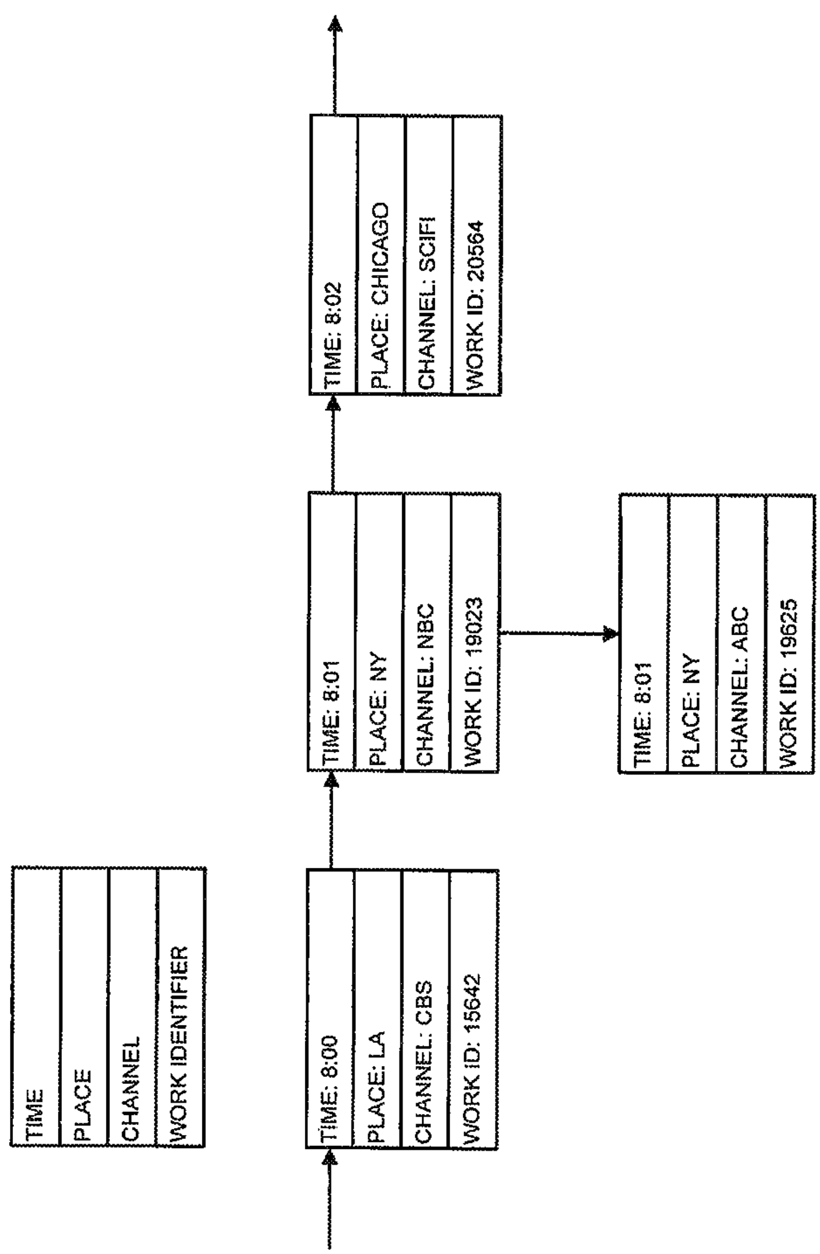

TIME
PLACE
CHANNEL
WORK IDENTIFIER
TIME: 8:00
PLACE: LA
CHANNEL: CBS
WORK ID: 15642
TIME: 8:01
PLACE: NY
CHANNEL: NBC
WORK ID: 19023
TIME: 8:01
PLACE: NY
CHANNEL: ABC
WORK ID: 19625
TIME: 8:02
PLACE: CHICAGO
CHANNEL: SCIFI
WORK ID: 20564

FIGURE 9

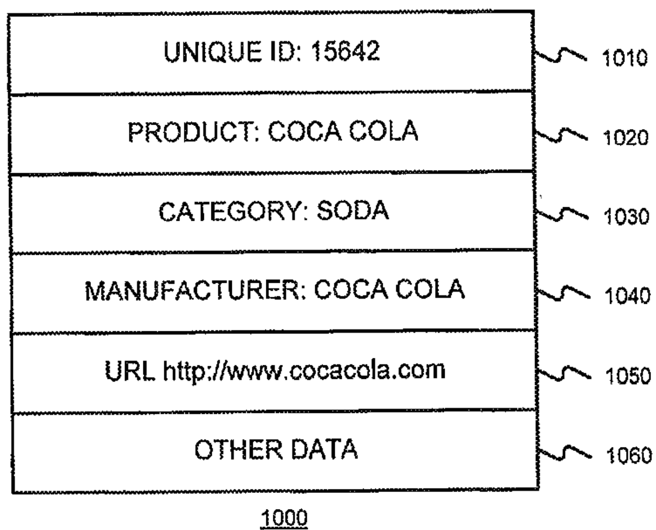

UNIQUE ID: 15642
PRODUCT: COCA COLA
CATEGORY: SODA
MANUFACTURER: COCA COLA
URL http://www.cocacola.com
OTHER DATA
1010
1020
1030
1040
1050
1060
1000

FIGURE 10

SYSTEM FOR USING EXTRACTED FEATURES FROM AN ELECTRONIC WORK

§0. RELATED APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 13/800,573 (incorporated herein by reference), titled "METHOD FOR TAGGING AN ELECTRONIC MEDIA WORK TO PERFORM AN ACTION," filed on Mar. 13, 2013, and listing Ingemar J. Cox as the inventor, which is a continuation of U.S. patent application Ser. No. 13/338,079 (incorporated herein by reference), titled "METHOD FOR USING EXTRACTED FEATURES FROM AN ELECTRONIC WORK," filed on Dec. 27, 2011, and listing Ingemar J. Cox as the inventor, which is a continuation of U.S. patent application Ser. No. 11/977,202 (incorporated herein by reference, issued as U.S. Pat. No. 8,205,237 on Jun. 19, 2012), titled "IDENTIFYING WORKS, USING A SUB-LINEAR TIME SEARCH, SUCH AS AN APPROXIMATE NEAREST NEIGHBOR SEARCH, FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET", filed Oct. 23, 2007, and listing Ingemar J. Cox as the inventor, which is a continuation of U.S. patent application Ser. No. 11/445,928 (incorporated herein by reference, issued as U.S. Pat. No. 8,010,988 on Aug. 30, 2011), titled "USING FEATURES EXTRACTED FROM AN AUDIO AND/OR VIDEO WORK TO OBTAIN INFORMATION ABOUT THE WORK," filed on Jun. 2, 2006, and listing Ingemar J. Cox as the inventor, which is a continuation-in-part of U.S. patent application Ser. No. 09/950,972 (incorporated herein by reference, issued as U.S. Pat. No. 7,058,223 on Jun. 6, 2006), titled "IDENTIFYING WORKS FOR INITIATING A WORK-BASED ACTION, SUCH AS AN ACTION ON THE INTERNET," filed on Sep. 13, 2001, and listing Ingemar J. Cox as the inventor, which application claims benefit to the filing date of provisional patent application Ser. No. 60/232,618 (incorporated herein by reference), titled "IDENTIFYING AND LINKING TELEVISION, AUDIO, PRINT AND OTHER MEDIA TO THE INTERNET", filed on Sep. 14, 2000 and listing Ingemar J. Cox as the inventor.

§1. BACKGROUND OF THE INVENTION

§1.1 Field of the Invention

The present invention concerns linking traditional media to new interactive media, such as that provided over the Internet for example. In particular, the present invention concerns identifying a work (e.g., content or an advertisement delivered via print media, or via a radio or television broadcast) without the need to modify the work.

§1.2 Related Art

§1.2.1 Opportunities Arising from Linking Works Delivered Via Some Traditional Media Channel or Conduit to a More Interactive System

The rapid adoption of the Internet and associated World Wide Web has recently spurred interest in linking works, delivered via traditional media channels or conduits, to a more interactive system, such as the Internet for example. Basically, such linking can be used to (a) promote commerce, such as e-commerce, and/or (b) enhance interest in the work itself by facilitating audience interaction or participation. Commerce opportunities include, for example, facilitating the placement of direct orders for products, providing product coupons, providing further information related to a product, product placement, etc.

In the context of e-commerce, viewers could request discount vouchers or coupons for viewed products that are redeemable at the point of purchase. E-commerce applications also extend beyond advertisements. It is now common for television shows to include product placements. For example, an actor might drink a Coke rather than a Pepsi brand of soda, actors and actresses might wear designer-labeled clothing such as Calvin Klein, etc. Viewers may wish to purchase similar clothing but may not necessarily be able to identify the designer or the particular style directly from the show. However, with an interactive capability, viewers would be able to discover this and other information by going to an associated Web site. The link to this Web site can be automatically enabled using the invention described herein.

In the context of facilitating audience interaction or participation, there is much interest in the convergence of television and computers. Convergence encompasses a very wide range of capabilities. Although a significant effort is being directed to video-on-demand applications, in which there is a unique video stream for each user of the service, as well as to transmitting video signals over the Internet, there is also interest in enhancing the television viewing experience. To this end, there have been a number of experiments with interactive television in which viewers can participate in a live broadcast. There are a variety of ways in which viewers can participate. For example, during game shows, users can answer the questions and their scores can be tabulated. In recent reality-based programming such as the ABC television game show, "Big Brother", viewers can vote on contestants who must leave the show, and be eliminated from the competition.

§1.2.2 Embedding Work Identifying Code or Signals within Works

Known techniques of linking works delivered via traditional media channels to a more interactive system typically require some type of code, used to identify the work, to be inserted into the work before it is delivered via such traditional media channels. Some examples of such inserted code include (i) signals inserted into the vertical blanking interval ("VBI") lines of a (e.g., NTSC) television signal, (ii) watermarks embedded into images, (iii) bar codes imposed on images, and (iv) tones embedded into music.

The common technical theme of these proposed implementations is the insertion of visible or invisible signals into the media that can be decoded by a computer. These signals can contain a variety of information. In its most direct form, the signal may directly encode the URL of the associated Web site. However, since the alphanumeric string has variable length and is not a particularly efficient coding, it is more common to encode a unique ID. The computer then accesses a database, which is usually proprietary, and matches the ID with the associated web address. This database can be considered a form of domain name server, similar to those already deployed for network addresses. However, in this case, the domain name server is proprietary and the addresses are unique ID's.

There are two principal advantages to encoding a proprietary identifier into content. First, as previously mentioned, it is a more efficient use of the available bandwidth and second, by directing all traffic to a single Web site that contains the database, a company can maintain control over the technology and gather useful statistics that may then be sold to advertisers and publishers.

As an example of inserting signals into the vertical blanking interval lines of a television signal, RespondTV of San Francisco, Calif. embeds identification information into the vertical blanking interval of the television signal. The VBI is part of the analog video broadcast that is not visible to television viewers. For digital television, it may be possible to encode the information in, for example, the motion picture experts group ("MPEG") header. In the USA, the vertical blanking interval is currently used to transmit close-captioning information as well as other information, while in the UK, the VBI is used to transmit teletext information. Although the close captioning information is guaranteed to be transmitted into the home in America, unfortunately, other information is not. This is because ownership of the vertical blanking interval is disputed by content owners, broadcasters and local television operators.

As an example of embedding watermarks into images, Digimarc of Tualatin, Oreg. embeds watermarks in print media. Invisible watermarks are newer than VBI insertion, and have the advantage of being independent of the method of broadcast. Thus, once the information is embedded, it should remain readable whether the video is transmitted in NTSC, PAL or SECAM analog formats or newer digital formats. It should be more reliable than using the vertical blanking interval in television applications. Unfortunately, however, watermarks still require modification of the broadcast signal which is problematic for a number of economic, logistical, legal (permission to alter the content is needed) and quality control (the content may be degraded by the addition of a watermark) reasons.

As an example of imposing bar codes on images, print advertisers are currently testing a technology that allows an advertisement to be shown to a camera, scanner or bar code reader that is connected to a personal computer ("PC"). The captured image is then analyzed to determine an associated Web site that the PC's browser then accesses. For example, GoCode of Draper, Utah embeds small two-dimensional bar codes for print advertisements. The latter signal is read by inexpensive barcode readers that can be connected to a PC. AirClic of Blue Bell, Pa. provides a combination of barcode and wireless communication to enable wireless shopping through print media. A so-called "CueCat" reads bar codes printed in conjunction with advertisements and articles in Forbes magazine. Similar capabilities are being tested for television and audio media.

Machine-readable bar codes are one example of a visible signal. The advantage of this technology is that it is very mature. However, the fact that the signal is visible is often considered a disadvantage since it may detract from the aesthetic of the work delivered via a traditional media channel or conduit.

As an example of embedding tones into music, Digital Convergence of Dallas, Tex. proposes to embed identification codes into audible music tones broadcast with television signals.

All the foregoing techniques of inserting code into a work can be categorized as active techniques in that they must alter the existing signal, whether it is music, print, television or other media, such that an identification code is also present. There are several disadvantages that active systems share. First, there are aesthetic or fidelity issues associated with bar codes, audible tones and watermarks. More importantly, all media must be processed, before it is delivered to the end user, to contain these active signals. Even if a system is enthusiastically adopted, the logistics involved with inserting bar codes or watermarks into, say every printed advertisement, are formidable.

Further, even if the rate of adoption is very rapid, it nevertheless remains true that during the early deployment of the system, most works will not be tagged. Thus, consumers that are early-adopters will find that most media is not identified. At best, this is frustrating. At worst, the naive user may conclude that the system is not reliable or does not work at all. This erroneous conclusion might have a very adverse effect on the adoption rate.

Further, not only must there be modification to the production process, but modifications must also be made to the equipment in a user's home. Again, using the example of watermarking of print media, a PC must be fitted with a camera and watermark detection software must be installed. In the case of television, the detection of the identification signal is likely to occur at the set-top-box—this is the equipment provided by the local cable television or satellite broadcasting company. In many cases, this may require modifications to the hardware, which is likely to be prohibitively expensive. For example, the audible tone used by Digital Convergence to recognize television content, must be fed directly into a sound card in a PC. This requires a physical connection between the television and the PC, which may be expensive or at least inconvenient, and a sound card may have to be purchased.

§1.2.3 Unmet Needs

In view of the foregoing disadvantages of inserting an identification code into a work, thereby altering the existing signal, there is a need for techniques of identifying a work without the need of inserting an identification code into a work. Such an identification code can then be used to invoke a work-related action, such as work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§2. SUMMARY OF THE INVENTION

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable programs for linking a media work to an action. Such embodiments might (a) extract features from the media work, (b) determine an identification of the media work based on the features extracted using a sub-linear time search, such as an approximate nearest neighbor search for example, and (c) determine an action based on the identification of the media work determined. In some embodiments consistent with the present invention, the media work is an audio signal. The audio signal might be obtained from a broadcast, or an audio file format. In other embodiments consistent with the present invention, the media work is a video signal. The video signal might be obtained from a broadcast, or a video file format.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§3. BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work.

FIG. 2 is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 3 is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 4 is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work.

FIG. 5 is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work.

FIG. 6 is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 7 is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work.

FIG. 8 is a block diagram illustrating an environment in which the present invention may operate.

FIG. 9 is an exemplary data structure in which extra-work information is associated with a work identifier.

FIG. 10 is an exemplary data structure including work-related actions.

§4. DETAILED DESCRIPTION

The present invention may involve novel methods, apparatus and data structures for identifying works without the need of embedding signals therein. Once identified, such information can be used to determine a work-related action. The following description is presented to enable one skilled in the art to make and use the invention, and is provided in the context of particular embodiments and methods. Various modifications to the disclosed embodiments and methods will be apparent to those skilled in the art, and the general principles set forth below may be applied to other embodiments, methods and applications. Thus, the present invention is not intended to be limited to the embodiments and methods shown and the inventors regard their invention as the following disclosed methods, apparatus, data structures and any other patentable subject matter to the extent that they are patentable.

§4.1 Functions

The present invention functions to identify a work without the need of inserting an identification code into a work. The present invention may do so by (i) extracting features from the work to define a feature vector, and (ii) comparing the feature vector to feature vectors associated with identified works. Alternatively, or in addition, the present invention may do so by (i) accepting extra-work information, such as the time of a query or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and (ii) use such extra-work information to lookup an identification of the work. In either case, an identification code may be used to identify the work.

The present invention may then function to use such an identification code to initiate a work-related action, such as for work-related commerce methods and/or to increase audience interest by facilitating audience interaction and/or participation.

§4.2 Embodiments

As just introduced in §4.1 above, the present invention may use intra-work information and/or extra-work information to identify a work. Once identified, such identification can be used to initiate an action, such as an action related to commerce, or facilitating audience participation or interaction. Exemplary embodiments of the present invention, in which work is recognized or identified based on intra-work information, are described in §4.2.1. Then, exemplary embodiments of the present invention, in which work is recognized or identified based on extra-work information, are described in §4.2.2.

§4.2.1 Embodiments in which Work is Recognized Based on Intra-Work Information, Such as a Feature Vector

Operations related to this embodiment are described in §4.2.1.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.1.2.

§4.2.1.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. 1 is a process bubble diagram of operations that may be performed in accordance with one version of the present invention, in which intra-work information is used to identify the work. As shown, a work-identification information storage 110 may include a number of items or records 112. Each item or record 112 may associate a feature vector of a work 114 with a, preferably unique, work identifier 116. The work-identification information storage 110 may be generated by a database generation operation(s) 120 which may, in turn, use a feature extraction operation(s) 122 to extract features from a work at a first time (WORK.sub.@t1), as well as a feature-to-work identification tagging operation(s) 124.

Further, work identifier-action information storage 130 may include a number of items or records 132. Each item or record 132 may associate a, preferably unique, work identifier 134 with associated information 136, such as an action for example. The work identifier-action information storage 130 may be generated by a database generation operation(s) 138 which may, for example, accept manual entries.

As can be appreciated from the foregoing, the work-information storage 110 records 112 and the work identification-action 130 records 132 can be combined into a single record. That is, there need not be two databases. A single database is also possible in which the work identifier, or a feature vector extracted from the work, serves as a key and the associated field contains work-related information, such as a URL for example.

The feature extraction operation(s) 140 can accept a work, such as that being rendered by a user, at a second time (WORK.sub.@t2), and extract features from that work. The extracted features may be used to define a so-called feature vector.

The extracted features, e.g., as a feature vector, can be used by a feature (vector) lookup operation(s) 150 to search for a matching feature vector 114. If a match, or a match within a predetermined threshold is determined, then the associated work identifier 116 is read.

The read work identifier can then be used by a work-associated information lookup operation(s) 160 to retrieve associated information, such as an action, 136 associated with the work identifier. Such information 136 can then be passed to action initiation operation(s) 170 which can perform some action based on the associated information 136.

§4.2.1.1.1 Exemplary Techniques for Feature Extraction

When the user initiates a request, the specific television or radio broadcast or printed commercial, each of which is referred to as a work, is first passed to the feature extraction operation. The work may be an image, an audio file or some portion of an audio signal or may be one or more frames or fields of a video signal, or a multimedia signal. The purpose of the feature extraction operation is to derive a compact representation of the work that can subsequently be used for the purpose of recognition. In the case of images and video, this feature vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n.times.n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, wavelet and or discrete cosine transforms. It might involve principal component analysis. It might also be a combination of these. For television and audio signals, recognition might also rely on a temporal sequence of feature vectors. The recognition literature contains many different representations. For block-based methods, blocks may be accessed at pseudo-random locations in each frame or might have a specific structure. For audio, common feature vectors are based on Fourier frequency decompositions, but other representations are possible. See, e.g., R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, 1973). See also K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Ed. (Academic Press, New York, 1990). (These references are incorporated herein by reference.)

As previously stated, one object of the vector extraction stage is to obtain a more concise representation of the frame. For example, each video frame is initially composed of 480.times.720 pixels which is equivalent to 345,600 pixels or 691,200 bytes. In comparison, an exemplary feature vector might only consist of 1 Kbyte of data.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e., horizontal and or vertical translation, or may undergo lossy compression such as by MPEG-2. It is advantageous that these and other processes do not adversely affect the extracted vectors. For still images there has been considerable work on determining image properties that are invariant to affine and other geometric distortions. For example, the use of Radon and Fourier-Mellin transforms have been proposed for robustness against rotation, scale and translation, since these transforms are either invariant or bare a simple relation to the geometric distortions. See, e.g., C. Lin, M. Wu, Y. M. Lui, J. A. Bloom, M. L. Miller, I. J. Cox, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," IEEE Transactions on Image Processing (2001). See also, U.S. Pat. Nos. 5,436,653, 5,504,518, 5,582,246, 5,612,729, and 5,621,454. (Each of these references is incorporated herein by reference.)

§4.2.1.1.2 Exemplary Techniques for Database Generation and Maintenance

A number of possibilities exist for generating and maintaining work identification (WID) and identification-action translation (WIDAT) databases. However, in all cases, works of interest are processed to extract a representative feature vector and this feature vector is assigned a unique identifier. This unique identifier is then entered into the work identification (WID) database **110** as well as into the WIDAT database **130** together with all the necessary associated data. This process is referred to as tagging. For example, in the case of an advertisement, the WIDAT database **130** might include the manufacturer (Ford), the product name (Taurus), a product category (automotive) and the URL associated with the Ford Taurus car together with the instruction to translate the query into the associated URL.

The determination of all works of interest and subsequent feature vector extraction and tagging depends on whether content owners are actively collaborating with the entity responsible for creating and maintaining the database. If there is no collaboration, then the database entity must collect all works of interest and process and tag them. While this is a significant effort, it is not overwhelming and is certainly commercially feasible. For example, competitive market research firms routinely tabulate all advertisements appearing in a very wide variety of print media. Newspapers and magazines can be scanned in and software algorithms can be applied to the images to identify likely advertisements. These possible advertisements can then be compared with advertisements already in the WID database **110**. If there is a match, nothing further need be done. If there is not a match, the image can be sent to a human to determine if the page does indeed contain an advertisement. If so, the operator can instruct the computer to extract the representative feature vector and assign it a unique identifier. Then, the operator can insert this information into the content identification database and as well as update the corresponding WIDAT database **130** with all the necessary associated data. This is continually performed as new magazines and papers include new advertisements to maintain the databases. This is a cost to the database entity. Television and radio broadcasts can also be monitored and, in fact, broadcast monitoring is currently performed by companies such as Nielsen Media research and Competitive Media Reporting. Television and radio broadcasts differ from print media in the real-time nature of the signals and the consequent desire for real-time recognition.

In many cases, advertisers, publishers and broadcasters may wish to collaborate with the database provider. In this case, feature extraction and annotation and/or extra-work information may be performed by the advertiser, advertisement agency, network and/or broadcaster and this information sent to the database provider to update the database. Clearly, this arrangement is preferable from the database provider's perspective. However, it is not essential.

§4.2.1.1.3. Exemplary Techniques for Matching Extracted Features with Database Entries

The extracted feature vector is then passed to a recognition (e.g., feature look-up) operation, during which, the vector is compared to entries of known vectors **114** in a content identification (WID) database **110**. It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search might not be possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used including mutual information, Euclidean distance and Lp-norms. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and false negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match. See, e.g., R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, 1973). See also, U.S. Pat. No. 3,919,474 by W. D. Moon, R. J. Weiner, R. A. Hansen and R. N. Linde, entitled "Broadcast Signal Identification System". (Each of these references is incorporated herein by reference.)

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. Unfortunately, binary search is not possible when taking a noisy signal and trying to find the most similar reference signal. This problem is one of nearest neighbor search in a (high-dimensional) feature space. In previous work, it was not uncommon to perform a linear search of all N entries, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this search can be computationally very expensive.

Other forms of matching include those based on clustering, kd-trees, vantage point trees and excluded middle vantage point forests are possible and will be discussed in more detail later. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, (Jan. 15, 1999). See also, P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370. (Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point.

If the extracted vector "matches" a known vector in the content identification database, then the work has been identified. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when the vector extracted from a work is not matched to the database even though the work is present in the database. There are several reasons why a works feature vector may fail to match a feature vector database entry. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will often contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible.

Finally, there is the case where the observed work is not present in the database. In this case, the work can be sent to an operator for identification and insertion in the database.

§4.2.1.1.4 Exemplary Work Based Actions

Assuming that the work is correctly identified, then the identifier can be used to retrieve associated information from the second work identification-action translation (WIDAT) database **130** that contains information **136** associated with the particular work **134**. This information may simply be a corresponding URL address, in which case, the action can be considered to be a form of network address translation. However, in general, any information about the work could be stored therein, together with possible actions to be taken such as initiating an e-commerce transaction. After looking up the work identifier **134** in the WIDAT database **130**, an action is performed on behalf of the user, examples of which has been previously described.

In addition to using the system to allow audience members of a work to connect to associated sites on the Internet, a number of other uses are possible. First, the work identification database **130** allows competitive market research data to be collected (e.g., the action may include logging an event). For example, it is possible to determine how many commercials the Coca Cola Company in the Chicago market aired in the month of June. This information is valuable to competitors such as Pepsi. Thus, any company that developed a system as described above could also expect to generate revenue from competitive market research data that it gathers.

Advertisers often wish to ensure that they receive the advertising time that was purchased. To do so, they often hire commercial verification services to verify that the advertisement or commercial did indeed run at the expected time. To do so, currently deployed systems by Nielsen and CMR embedded active signals in the advertisement prior to the broadcast. These signals are then detected by remote monitoring facilities that then report back to a central system which commercials were positively identified. See for example U.S. Pat. Nos. 5,629,739 by R. A. Dougherty entitled "Apparatus and method for injecting an ancillary signal into a low energy density portion of a color television frequency spectrum", 4,025,851 by D. E. Haselwood and C. M. Solar entitled "Automatic monitor for programs broadcast", 5,243,423 by J. P. DeJean, D. Lu and R. Weissman, entitled "Spread spectrum digital data transmission over TV video", and 5,450,122 by L. D. Keene entitled "In-station television program encoding and monitoring system and method". (Each of these patents is incorporated herein by reference.) Active systems are usually preferred for advertisement verification because the required recognition accuracy is difficult to achieve with passive systems. The passive monitoring system described herein supports commercial verification.

§4.2.1.2 Exemplary Architectures

Three alternative architectural embodiments in which the first technique may be employed are now described with reference to FIGS. **2**, **3**, and **4**.

FIG. **2** is a block diagram illustrating a first embodiment of the present invention, in which intra-work information is used to identify the work and in which a audience member device **210**, such as a PC for example, receives and renders a work that is consumed by an audience member (user). At some point, the user may wish to perform a work-specific action such as traversing to an associated Web site. Upon initiation of this request, the computer **210** performs the operations **140***a*, **150***a*, **160***a* and **170***a*, such as those shown in FIG. **1**. To reiterate, these operations include a feature extraction operation(s) **140***a*, feature vector lookup or matching operation(s) **150***a* in connection with items or records **112***a* in a work-identification (WID) database **110***a*. If a matching feature vector **114***a* is found, the work-associated information lookup operation(s) **160***a* can use the associated work identifier **116***a* to accessing a work identification-action translation (WIDAT) database **130***a* to retrieve associated information **136***a*, possibly including determining what action should be performed.

As described above, the two databases might be integrated into a single database. However, conceptually, they are described here as separate.

An example illustrating operations that can occur in the first embodiment of FIG. **1**, is now described. Consider a print application, in which say 10,000 advertisements are to be recognized that appear in national newspapers and magazines. If 1 Kbyte is required to store each feature vector then approximately 10 Mbytes of storage will be required for the work identification database **110***a*. Such a size does not represent a serious problem, in either memory or disk space, to present personal computers.

An important issue then becomes recognition rate. While this may be problematic, all the images are two-dimensional-three-dimensional object recognition is not required. Of course, since a low cost camera captures the printed advertisement, there may be a number of geometric distortions that might be introduced together with noise. Nevertheless, the application is sufficiently constrained that adequate recognition rates should be achievable with current state-of-the-art computer vision algorithms. See, e.g., P. N. Yianilos "Excluded Middle Vantage Point Forests for nearest Neighbor Search", Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, Jan. 15, 1999. See also, P. N. Yianilos "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370.

(Each of these references is incorporated herein by reference.) Thus, for example, a sub-linear search time can be achieved. Unlike the kd-tree method which finds the nearest neighbor with certainty, randomized constructions, like the one described in P. N. Yianilos, "Locally lifting the curse of Dimensionality for nearest Neighbor Search" SODA 2000: 361-370, that succeed with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. Estimates of the size of the WIDAT database **130***a* depend on what associated information (recall fields **136**) is stored. If, for example, only a URL address is needed, about 20 characters can typically represent most URLs. Thus, the size of the WIDAT database **130***a* would be less than 1 Mbyte.

The configuration just described with reference to FIG. **2** places all of the processing and data on each user's local machine **210**. A number of alternative embodiments, in which some or all of the storage and processing requirements are performed remotely, will be described shortly.

As new works are created and made publicly available, the databases residing on a user's local computer become obsolete. Just as the database provider **240** must continually update the databases in order to remain current, there is also a need to update local databases on devices at audience member premises. This update process can be performed over the Internet **230** in a manner very similar to how software is currently upgraded. It is not necessary to download an entirely new database although this is an option. Rather, only the changes need to be transmitted. During this update process, the user's computer **210** might also transmit information to a central monitoring center **240** informing it of which advertisements the computer user has queried. This type of information is valuable to both advertisers and publishers. Of course, care must be taken to ensure the privacy of individual users of the system. However, it is not necessary to know the identity of individual users for the system to work.

FIG. **3** is a block diagram illustrating a second embodiment of the present invention, in which intra-work information is used to identify the work. Although the WIDAT database can be quite small, as illustrated in the exemplary embodiment described above with respect to FIG. **2**, there is still the problem of keeping this database current. While periodic updates of the local databases may be acceptable, they become unnecessary if the WIDAT database **130***b* is at a remote location **340**. In this arrangement, illustrated in FIG. **3**, after the local computer **310** identifies the work, it sends a query to the remote WIDAT database **130***b*. The query may contain the work identifier. The remote site **340** may then return the associated information **136**. Although the remote WIDAT database **130***b* needs to be updated by the database provider, this can be done very frequently without the need for communicating the updates to the local computers **310**.

The second embodiment is most similar to active systems in which an embedded signal is extracted and decoded and the identifier is used to interrogate a central database. Consequently it has many of the advantages of such systems, while avoiding the need to insert signals into all works. One such advantage, is that the database provider receives real-time information relating to users' access patterns.

The WIDAT database **130***b* might physically reside at more than one location. In such a case, some requests will go to one site, and other requests will go to another. In this way, overloading of a single site by too many users can be avoided. Other load balancing techniques are also applicable.

FIG. **4** is a block diagram illustrating a third embodiment of the present invention, in which intra-work information is used to identify the work. Recall that the WIDAT database may be small relative to that work identification database (WID). As the size of the work recognition (WID) database increases, the foregoing embodiments may become impractical. Consider, for example, a music application in which it is desired to identify 100,000 song titles. If it is again assumed that a 1 Kbyte vector can uniquely represent each song, then on the order of 100 Mbytes is now needed. This size is comparable to large application programs such as Microsoft's Office 2000 suite. Although this still does not represent an inordinate amount of disk space, if this data needs to reside in memory at all times, then very few present machines will have adequate resources. Clearly, at some point, the proposed architectures scales to a point where requirements become impractical. In this case, a further modification to the architecture is possible.

Since the storage and searching of the work-identifier (WID) database require the most computation and storage, it may be more economical to perform these actions remotely. Thus, for example, if a user is playing an MP3 music file and wants to go to a corresponding website, the MP3 file is passed to an operation that determines one or more feature vectors. In the third embodiment, instead of performing the matching locally **410**, the one or more vectors are transmitted to a central site **440** at which is stored the WID and WIDAT databases **110***c* and **130***c* together with sufficiently powerful computers to resolve this request and those of other computer users. This configuration is illustrated in FIG. **4**. Similarly, if a user is playing an MPEG or other video file and wants to initiate a work-related action, the video file is passed to an operation **140***c* that extracts one or more feature vectors. The entire video file need not be processed. Rather, it may be sufficient to process only those frames in the temporal vicinity to the users request, i.e., to process the current frame and or some number of frames before and after the current frame, e.g. perhaps 100 frames in all. The extracted feature vector or feature vectors can then be transmitted to a central site **440** which can resolve the request.

After successfully matching the feature vector, the central site **440** can provide the user with information directly, or can direct the user to another Web site that contains the information the user wants. In cases where the recognition is ambiguous, the central site **440** might return information identifying one of several possible matches and allow the user to select the intended one.

The third embodiment is particularly attractive if the cost of extracting the feature vector is small. In this case, it becomes economical to have feature vector extraction **140***c* in digital set-top-boxes and in video recorders **410**. The latter may be especially useful for the new generation of consumer digital video recorders such as those manufactured by TIVO and Replay TV. These devices already have access to the Internet via a phone line. Thus, when someone watching a recorded movie from television reacts to an advertisement, the video recorder would extract one or more feature vectors and transmit them to a central site **440**. This site **440** would determine if a match existed between the query vector and the database of pre-stored vectors **110***c*. If a match is found, the central server **440** would transmit the associated information, which might include a Web site address or an 800 number for more traditional ordering, back to the audience user device **410**. Of course, a consumer device **410** such as a digital video recorder might also store personal information of the owner to facilitate online e-commerce. Such a device **410** could store the owner's name, address, and credit card information and automatically transmit them to an on-line store to complete a purchase. Very little user interaction other than to authorize the purchase might be needed. This type of purchasing may be very convenient to consumers.

Another advantage of the third embodiment is that it obviates the need to update local databases while, at the same time, the centrally maintained databases can be kept current with very frequent updating.

§4.2.2 Embodiments in which Work is Recognized Based on Extra-Work Information

Operations related to this embodiment are described in §4.2.2.1 below. Then, various architectures which may be used to effect such operations are described in §4.2.2.2.

If the cost of extracting a feature vector is too large, then the cost of deploying any of the embodiments described in §4.2.1 above may be prohibitive. This is particularly likely in very cost sensitive consumer products, including set-top-boxes and next generation digital VCRs. Acknowledging this fact, a different technique, one that is particularly well suited for broadcasted media such as television and radio as well as to content published in magazines and newspapers, is now described. This technique relies on the fact that a work need not be identified by a feature vector extracted from the work (which is an example of "intra-work information"), but can also be identified by when and where it is published or broadcast (which are examples of "extra-work information")

An example serves to illustrate this point. Consider the scenario in which a viewer sees a television commercial and responds to it. The embodiments described in §4.2.1 above required the user device (e.g., a computer or set-top-box) **210/310/410** to extract a feature vector. Such an extracted vector was attempted to be matched to another feature vector(s), either locally, or at a remote site. In the embodiments using a remote site, if the central site is monitoring all television broadcasts, then the user's query does not need to include the feature vector. Instead, the query simply needs to identify the time, geographic location and the station that the viewer is watching. A central site can then determine which advertisement was airing at that moment and, once again, return the associated information. The same is true for radio broadcasts. Moreover, magazines and newspapers can also be handled in this manner. Here the query might include the name of the magazine, the month of publication and the page number.

§4.2.2.1 Operations and Exemplary Methods and Techniques for Effecting Such Operations

FIG. **5** is a process bubble diagram of operations that may be performed in accordance with another version of the present invention, in which extra-work information is used to identify the work. As shown, a query work-identification (QWID) information storage **510** may include a number of items or records **512**. Each item or record **512** may associate extra-work information **514**, related to the work, with a, preferably unique, work identifier **516**. The query work-identification (QWID) information storage **510** may be generated by a database generation operation(s) **520**.

Further, work identifier-action information (WIDAT) storage **530** may include a number of items or records **532**. Each item or record **532** may associate a, preferably unique, work identifier **534** with associated information **536**, such as an action for example. The work identifier-action (WIDAT) information storage **530** may be generated by a database generation operation(s) **538** which may, for example, accept manual entries.

As can be appreciated from the foregoing, the query work-information (QWID) storage **510** records **512** and the work identification-action (WIDAT) storage **530** records **532** can be combined into a single record.

The extra-work information aggregation (e.g., query generation) operation(s) **540** can accept a information related to a work, such as the time of a user request or of a rendering of the work, the geographic location at which the work is rendered, and the station that the audience member has selected, and generate a query from such extra-work information.

The query including the extra-work information can be used by a lookup operation(s) **550** to search for a "matching" set of information **514**. If a match, or a match within a predetermined threshold is determined, then the associated work identifier **516** is read.

The read work identifier can then be used by a work-associated information lookup operation(s) **560** to retrieve associated information, such as an action, **536** associated with the work identifier. Such information **536** can then be passed to action initiation operation(s) **570** which can perform some action based on the associated information **536**.

If the extra-work information of a work is known (in advance), generating the query work identifier (QWID) information **510** is straight-forward. If this were always the case, an intra-work information-based recognition operation would not be needed. However, very often this is not the case. For example, local television broadcasts typically have discretion to insert local advertising, as well as national advertising. Thus, it often is not possible to know in advance when, on what station, and where a particular advertisement will play.

In such instances, a real-time (e.g., centralized) monitoring facility **580** may be used to (i) extract feature vectors from a work, (ii) determine a work identifier **116** from the extracted features, and (iii) communicate one or more messages **590** in which extra-work information (e.g., time, channel, geographic market) **592** is associated with a work identifier **594**, to operation(s) **520** for generating query work identification (QWID) information **510**.

§4.2.2.1.1 Exemplary Extra-Work Information

In the context of national broadcasts, geographic information may be needed to distinguish between, for example, the ABC television broadcast in Los Angeles and that in New York. While both locations broadcast ABC's programming, this programming airs at different times on the East and West coasts of America. More importantly, the local network affiliates that air ABC's shows have discretion to sell local advertising as well as a responsibility to broadcast the national commercials that ABC sells. In short, the works broadcast by ABC in Los Angeles can be different from that in other geographic locations. Geographic information is therefore useful to distinguish between the different television markets. In some circumstances, geographic information may not be necessary, especially in parts of the world with highly regulated and centralized broadcasting in which there are not regional differences.

§4.2.2.1.2 Exemplary Techniques for Generating Databases

FIG. **5** illustrates a third database **510** referred to as the query to work identification (QWID) database. This database **510** maps the query (e.g., in the form of time, location and channel information) into a unique ID that identifies the perceived work. The QWID **510** and WIDAT **530** databases might not be separate, but for clarity will be considered so. After retrieving the unique work identifier **512** from the QWID database **510**, the identifier can be used to access the WIDAT database **530**. This is discussed in more detail later.

As introduced above, although it appears that this architecture does not require a recognition facility, such a facility may be needed. The feature extraction operation(s) **140***d*, as well as the work identification operation(s) **150***d* and other databases **110***d*, may be moved to one or more remote sites **580**.

Although TV Guide and other companies provide detailed information regarding what will be broadcast when, these scheduling guides do not have any information regarding what advertisements will air when. In many cases, this information is unknown until a day or so before the broadcast. Even then, the time slots that a broadcaster sells to an advertiser only provide a time range, e.g. 12 pm to 3 pm. Thus it is unlikely that all commercials and aired programming can be determined from TV schedules and other sources prior to transmission. Further, occasionally programming schedules are altered unexpectedly due to live broadcasts that overrun their time slots. This is common in sports events and awards shows. Another example of interrupts to scheduled programming occurs when a particularly important news event occurs.

During transmission, it may therefore be necessary for a central site **580** to determine what work is being broadcast and to update its and/or other's database **520** accordingly based on the work identified **594** and relevant extra-work information **592**. There are a variety of ways that this can be accomplished.

First, it may be economically feasible to manually monitor all television stations that are of interest, and manually update the database with information regarding the work being monitored. In fact, Nielsen used such procedures in the early 1960's for the company to tabulate competitive market data. More than one person can be employed to watch the same channel in order to reduce the error rate. It should be noted that the recent ruling by the FCC that satellite broadcasters such as DirecTV, DishTV and EchoStar can carry local stations significantly reduces the cost of monitoring many geographic markets. Currently, DirecTV, for example, carries the four main local stations in each of the 35 largest markets. Thus, these 4.times.35=140 channels can all be monitored from a single site **580**. This site would be provided with satellite receivers to obtain the television channels.

Unfortunately, however, humans are error prone and the monitoring of many different stations from many different geographic locations can be expensive. In order to automate the recognition process, a central site **580** could employ a computer-based system to perform automatic recognition. Because the recognition is centralized, only one or a few sites are needed. This is in comparison with the first architecture we described in which a complete recognition system was required in every user's home or premise. This centralization makes it more economic to employ more expensive computers, perhaps even special purpose hardware, and more sophisticated software algorithms. When video frames or clips cannot be identified or are considered ambiguous, this video can be quickly passed to human viewers to identify. Further, it should be possible for the automated recognition system to use additional information such as television schedules, time of day, etc in order to improve its recognition rate.

§4.2.2.1.2 Exemplary Techniques for Generating Queries Based on Extra-Work Information

At the audience member (user) premises, all that is needed is for the device to send a query to a database-server with information that includes extra-work information, such as geographic location, time and channel. Usually, this extra-work information would be transmitted in real-time, while the work (e.g., an advertisement) is being broadcast. However, this is not necessary. If the television does not have access to the Internet, and most TVs do not yet, then an audience member (user) may simply remember or record which channel he or she was viewing at what time. In fact, the user device could store this information for later retrieval by the user. At a convenient later time, the user might access the Internet using a home PC. At this time, he or she can query the database by entering this extra-work information (e.g., together with geographic information) into an application program or a web browser plug-in.

Another possibility is allowing an audience member (user), at the time he or she is consuming (e.g., viewing, reading, listening to, etc.) the work, to enter query information into a handheld personal digital assistant ("PDA") such as a Palm Pilot, so as not to forget it. This information can then be manually transferred to a device connected to a network, or the information can be transferred automatically using, for example, infrared communications or via a physical link such as a cradle. Recently, PDAs also have some wireless networking capabilities built in, and thus might support direct access to the information desired. Further, software is available that allows a Palm Pilot or other PDA to function as a TV remote control device. As such, the PDA already knows the time of day and channel being viewed. It also probably knows the location of the audience member, since most PDA users include their own name and address in the PDA's phonebook and identify it as their own. Thus, with one or a few clicks, an audience member PDA user could bookmark the television content he or she is viewing. If the PDA is networked, then the PDA can, itself, retrieve the associated information immediately. Otherwise, the PDA can transfer this bookmarked data to a networked device, which can then provide access to the central database.

§4.2.2.2 Exemplary Architectures

FIG. **6** is a block diagram illustrating a fourth embodiment of the present invention, in which extra-work information is used to identify the work. As shown, an extra-work information aggregation operation **540***a* may be effected on a device **610**, such as a PC, at the audience member (user) premises. The various databases **510***a*, **530***a*, and **110***e*, as well as the database generation operation(s) **520***a*/**538***a*, the lookup operation(s) **550***a* and the work-associated information lookup operation(s) **560***a* may be provided at one or more centralized monitoring and query resolution centers **640**.

FIG. **7** is a block diagram illustrating a fifth embodiment of the present invention, in which extra-work information is used to identify the work. This fifth embodiment is similar to the fourth embodiment illustrated in FIG. **6** but here, the monitoring center **740***a* and query resolution center **740***b* are separate.

These embodiments have many advantages for television and radio broadcasters who desire to provide Internet links or other action. First, the audience member (user) equipment, whether it is a computer, set-top-box, television, radio, remote control, personal digital assistant (pda), cell phone or other device, does not need to perform any processing of the received signal. As such, there is almost no cost involved to equipment manufacturers.

These last embodiments have some similarity with services such as those provided by the companies Real Names of Redwood City, Calif., America Online ("AOL") and especially iTag from Xenote. The popular press has reported on the difficulties associated with assigning domain names. The simplest of these problems is that almost all the one-word names in the ".com" category have been used. Consequently, domain names can often be difficult to remember. To alleviate this problem, RealNames and AOL provide alternative, proprietary name spaces (AOL calls these keywords). For a fee, a company may register a name with these companies. Thus, rather than type the URL, http://www.bell-labs.com, the simple keyword "bell" might be sufficient to access the same Web site. These capabilities are convenient to users. However, these systems are very different from the fourth and fifth embodiments described. First, and foremost, these systems are not designed to identify content. Rather, they are simply alternative network address translation systems based on easily remembered mnemonics which are sold to interested companies. As such, the user is still expected to type in an address, but this address is easier to remember than the equivalent URL. In contrast, while a user may manually enter the information describing the work, the preferred embodiment is for the computer, set-top-box or other device to automatically generate this information. Further, the mapping of keywords to network addresses is an arbitrary mapping maintained by AOL or Real Names. For example, the keyword "bell" might just as reasonably point to the Web site for Philadelphia's Liberty Bell as to Lucent's Bell Labs. In contrast, the query used in the fourth and fifth embodiments is designed to contain all the necessary data to identify the work, e.g. the time, place and television channel during which the work was broadcast. There is nothing arbitrary about this mapping. It should also be pointed out that the proposed system is dynamic—the same work, e.g. a commercial, potentially has an infinite number of addresses depending on when and where it is broadcast. If an advertisement airs 100,000 unique times, then there are 100,000 different queries that uniquely identify it. Moreover, the exemplary query includes naturally occurring information such as time, place, channel or page number. This is not the case for AOL or RealNames, which typically assigns one or more static keywords to the address of a Web site.

Xenote's iTag system is designed to identify radio broadcasts and uses a query similar to that which may be used in the fourth and fifth embodiments, i.e. time and station information. However, the work identification information is not dynamically constructed but is instead based on detailed program scheduling that radio stations must provide it. As such, it suffers from potential errors in scheduling and requires the detailed cooperation of broadcasters. While the fourth and fifth embodiments might choose to use program scheduling information and other ancillary information to aid in the recognition process, they do not exclusively rely on this. The concept of resolving a site name by recognizing the content is absent from the above systems.

§4.2.3 Exemplary Apparatus for Audience Member (User) Premise Device

While personal computers may be the primary computational device at a user's location, it is not essential to use a PC. This is especially true of the embodiments depicted in FIGS. 6 and 7, which do not require the content, e.g. video signal, to be processed. Instead, only a unique set of identification parameters such as time, location and channel are provided to identify the perceived Work. Many forms of devices can therefore take advantage of this configuration.

As previously noted, personal digital assistants (PDAs) can be used to record the identification information. This information can then be transferred to a device with a network communication such as a PC. However, increasingly, PDAs will already have wireless network communication capabilities built-in, as with the Palm VII PDA. These devices will allow immediate communication with the query resolution center and all information will be downloaded to them or they can participate in facilitating an e-commerce transaction. Similarly, wireless telephones are increasingly offering web-enabled capabilities. Consequently, wireless phones could be programmed to act as a user interface.

New devices can also be envisaged, including a universal remote control for home entertainment systems with a LCD or other graphical display and a network connection. This connection may be wireless or the remote control might have a phone jack that allows it to be plugged directly into an existing phone line. As home networks begin to be deployed, such devices can be expected to communicate via an inexpensive interface to the home network and from there to access the Internet.

In many homes, it is not uncommon for a computer and television to be used simultaneously, perhaps in the same room. A person watching television could install a web browser plug-in or applet that would ask the user to identify his location and the station being watched. Then, periodically, every 20 seconds for example, the plug-in would update a list of web addresses that are relevant to the television programs being watched, including the commercials. The audience member would then simply click on the web address of interest to obtain further information. This has the advantage that the viewer does not have to guess the relevant address associated with a commercial and, in fact, can be directed to a more specialized address, such as www.fordvehicles.com/ibv/tausrus2kflash/flash.html, rather than the generic www.ford.com site. Of course, this applet or plug-in could also provide the database entity with information regarding what is being accessed from where and at what time. This information, as noted earlier, is valuable to advertisers and broadcasters. For PC's that have infra-red communication capabilities, it is straightforward to either control the home entertainment center from the PC or for the PC to decode the signals from a conventional remote control. Thus, as a user changes channels, the PC is able to automatically track the channel changes.

Recording devices such as analog VCRs and newer digital recording devices can also be exploited in the embodiments depicted in FIGS. 6 and 7, especially if device also record the channel and time information for the recorded content. When a user initiates a query, the recorded time and channel, rather than the current time and channel, then form part of the identification information.

Digital set-top-boxes are also expected to exploit the capabilities described herein. In particular, such devices will have two-way communication capabilities and may even include cable modem capabilities of course, the two-way communication need not be over a television cable. For example, satellite set-top-boxes provide up-link communications via a telephone connection. Clearly, such devices provide a convenient location to enable the services described herein. Moreover, such services can be provided as part of the OpenCable and DOCSIS (data over cable service interface specification) initiatives.

§4.2.4 Information Retrieval Using Features Extracted from Audio and/or Video Works

Some embodiments consistent with the present invention provide a computer-implemented method, apparatus, or computer-executable program for providing information about an audio file or (a video file) played on a device. Such embodiments might (a) extract features from the audio (or video) file, (b) communicate the features to a database, and (c) receive the information about the audio (or video) file from the database. In some embodiments consistent with the present invention, the act of extracting the features is performed by a microprocessor of the device, and/or a digital signal processor of the device. The received information might be rendered on an output (e.g., a monitor, a speaker, etc.) of the device. The received information might be stored (e.g., persistently) locally on the device. The information might be stored on a disk, or non-volatile memory.

In some of the embodiments pertaining to audio files, the audio file might be an mp3 file or some other digital representation of an audio signal. The information might include a song title, an album title, and/or a performer name.

In some of the embodiments pertaining to video files, the video file might be an MPEG file or some other digital representation of a video signal. The video file might be a video work, and the information might include a title of the video work, a director of the video work, and names of performers in the video work.

§4.3 Operational Examples

An example illustrating operations of an exemplary embodiment of the present invention, that uses intra-work information to identify the work, is provided in §

4.3.1. Then, an example illustrating operations of an exemplary embodiment of the present invention, that uses extra-work information to identify the work, is provided in §4.3.2.

§4.3.1 Operational Example where Intra-Work Information is Used to Identify the Work

A generic system for monitoring television commercials is now described. Obviously, the basic ideas extend beyond this specific application.

The process of recognition usually begins by recognizing the start of a commercial. This can be accomplished by looking for black video frames before and after a commercial. If a number of black frames are detected and subsequently a similar number are detected 30 seconds later, then there is a good chance that a commercial has aired and that others will follow. It is also well known than the average sound volume during commercials is higher than that for television shows and this too can be used as an indicator of a commercial. Other methods can also be used. The need to recognize the beginning of a commercial is not essential. However, without this stage, all television programming must be assumed to be commercials. As such, all video frames must be analyzed. The advantage of determining the presence of a commercial is that less video content must be processed. Since the percentage of advertising time is relatively small, this can lead to considerable savings. For example, commercials can be buffered and then subsequently processed while the television show is being broadcast. This reduces the real-time requirements of a system at the expense of buffering, which requires memory or disk space. Of course, for the applications envisioned herein, a real-time response to a user requires real-time processing.

Once it is determined that an advertisement is being broadcast, it is necessary to analyze the video frames. Typically, a compact representation of each frame is extracted. This vector might be a pseudo-random sample of pixels from the frame or a low-resolution copy of the frame or the average intensities of n.times.n blocks of pixels. It might also be a frequency-based decomposition of the signal, such as produced by the Fourier, Fourier-Mellin, wavelet and or discrete cosine transforms. It might involve principal component analysis or any combination thereof. The recognition literature contains many different representations. For block-based methods, the n.times.n blocks may be located at pseudo-random locations in each frame or might have a specific structure, e.g. a complete tiling of the frame. The feature vector might then be composed of the pixels in each block or some property of each block, e.g. the average intensity or a Fourier or other decomposition of the block. The object of the vector extraction stage is to obtain a more concise representation of the frame. Each frame is initially composed of 480.times.720 pixels which is equivalent to 345,600 bytes, assuming one byte per pixel. In comparison, the feature vector might only consist of 1 Kbyte of data. For example, if each frame is completely tiled with 16.times.16 blocks, then the number of blocks per frame is 345,600/256=1350. If the average intensity of each block constitutes the feature vector, then the feature vector consists of 1350 bytes, assuming 8-bit precision for the average intensity values. Alternatively, 100 16.times.16 blocks can be pseudo-randomly located on each frame of the video. For each of these 100 blocks, the first 10 DCT coefficients can be determined. The feature vector then consists of the 100.times.10=1000 DCT coefficients. Many other variations are also possible. In many media applications, the content possesses strong temporal and spatial correlations. If necessary, these correlations can be eliminated or substantially reduced by pre-processing the content with a whitening filter.

A second purpose of the feature extraction process is to acquire a representation that is robust or invariant to possible noise or distortions that a signal might experience. For example, frames of a television broadcast may experience a small amount of jitter, i.e. horizontal and or vertical translation, or may undergo lossy compression such as MPEG-2. It is advantageous, though not essential, that these and other processes do not adversely affect the extracted vectors.

Each frame's feature vector is then compared with a database of known feature vectors. These known vectors have previously been entered into a content recognition database together with a unique identifier. If a frame's vector matches a known vector, then the commercial is recognized. Of course, there is the risk that the match is incorrect. This type of error is known as a false positive. The false positive rate can be reduced to any desired value, but at the expense of the false negative rate. A false negative occurs when a frame's vector is not matched to the database even though the advertisement is present in the database. There are several reasons why a frame's feature vector may fail to match. First, the recognition system may not be capable of 100% accuracy. Second, the extracted vector will contain noise as a result of the transmission process. This noise may alter the values of a feature vector to the extent that a match is no longer possible. Finally, there is the case where the observed commercial is not yet present in the database. In this case, it is necessary to store the commercial and pass it (e.g., to a person) for identification and subsequent entry in the database.

It is important to realize that the matching of extracted and known vectors is not equivalent to looking up a word in an electronic dictionary. Since the extracted vectors contain noise or distortions, binary search is often not possible. Instead, a statistical comparison is often made between an extracted vector and each stored vector. Common statistical measures include linear correlation and related measures such as correlation coefficient, but other methods can also be used, including clustering techniques. See, e.g., the Duda and Hart reference. These measures provide a statistical measure of the confidence of the match. A threshold can be established, usually based on the required false positive and negative rates, such that if the correlation output exceeds this threshold, then the extracted and known vectors are said to match.

If binary search was possible, then a database containing N vectors would require at most log(N) comparisons. However, in current advertisement monitoring applications there is no discussion of efficient search methods. Thus, a linear search of all N entries may be performed, perhaps halting the search when the first match is found. On average, this will require N/2 comparisons. If N is large, this can be computationally expensive. Consider a situation in which one out of 100,000 possible commercials is to be identified. Each 30-second commercial consists of 900 video frames. If all 900 frames are stored in the database, then N=90,000,000. Even if only every 10.sup.th video frame is stored in the database, its size is still nine million. While databases of this size are now common, they rely of efficient search to access entries, i.e., they do not perform a linear search. A binary search of a 90,000,000-item database requires less than 20 comparisons. In contrast, a linear search will require an average of 45,000,000!

With 9 million entries, if each vector is 1 Kbyte, then the storage requirement is 9 Gigabytes. Disk drives with this capacity are extremely cheap at this time. However, if the database must reside in memory due to real-time requirements, then this still represents a substantial memory requirement by today's standards. One reason that the data may need to be stored in memory is because of the real-time requirements of the database. If 10 channels are being simultaneously monitored within each of 50 geographic areas, then there will be 15,000 queries per second to the content recognition database, assuming each and every frame is analyzed. This query rate is low. However, if a linear search is performed then 675 billion comparisons per second will be required. This is an extremely high computational rate by today's standards. Even if only key frames are analyzed, this is unlikely to reduce the computational rate by more than an order of magnitude.

If an advertisement is not recognized, then typically, the remote monitoring system will compress the video and transmit it back to a central office. Here, the clip is identified and added to the database and the remote recognition sites are subsequently updated. Identification and annotation may be performed manually. However, automatic annotation is also possible using optical character recognition software on each frame of video, speech recognition software, close captioning information and other information sources. As these methods improve in accuracy, it is expected that they will replace manual identification and annotation.

The recognition system described can be considered to be a form of nearest neighbor search in a high dimensional feature space. This problem has been very well studied and is known to be very difficult as the dimensionality of the vectors increases. A number of possible data structures are applicable including kd-trees and vantage point trees. These data structures and associated search algorithms organize a N-point dataset (N=90,000,000 in out previous example) so that sub-linear time searches can be performed on average. However, worst-case search times can be considerably longer. Recently, Yianilos proposed an excluded middle vantage point forest for nearest neighbor search. See, e.g., the Yianilos reference. This data structure guarantees sub-linear worst-case search times, but where the search is now for a nearest neighbor within a fixed radius, .tau. The fixed radius search means that if the database contains a vector that is within .tau. of the query, then there is a match. Otherwise, no match is found. In contrast, traditional vantage point trees will always return a nearest neighbor, even if the distance between the neighbor and the query is very large. In these cases, if the distance between the query and the nearest neighbor exceeds a threshold, then they are considered not to match. This is precisely what the excluded middle vantage point forest implicitly does.

Using an excluded middle vantage point forest, will allow accurate real-time recognition of 100,000 broadcasted advertisements. This entails constructing an excluded middle vantage point forest based on feature vectors extracted from say 90,000,000 frames of video. Of course, using some form of pre-filtering that eliminates a large number of redundant frames or frames that are not considered to be good unique identifiers can reduce this number. One such pre-filter would be to only examine the I-frames used when applying MPEG compression. However, this is unlikely to reduce the work identification database (WID) size by more than one order of magnitude. Assuming 10 channels are monitored in each of 50 geographic regions, then the query rate is 15,000=10.times.50.times.30 queries per second.

§4.3.2 Operational Example where Extra-Work Information is Used to Identify the Work

FIG. 8 depicts a satellite television broadcast system **800**, though cable and traditional broadcast modes are also applicable. Block **810** represents audience members (users) watching a TV channel in their home, which also has a connection **812** to the Internet **820**. Other networks are also possible. The satellite broadcasts are also being monitored by one or more television monitoring centers **840*a***. These centers **840*a*** may monitor all or a subset of the television channels being broadcast. They are not restricted to monitoring satellite TV broadcasts but may also monitor cable and traditional terrestrial broadcasts. The primary purpose of these monitoring centers **840*a*** is to identify the works being broadcasted. Of particular interest are television advertisements. However, other works, or portions thereof, may also be identified. Each time a new segment of a work is identified, the monitoring system or systems **840*a*** update one or more database centers **840*b***, informing them of the time, place, channel and identity of the identified segment. The segment may be a complete thirty second commercial or, more likely, updates will occur more frequently, perhaps at a rate of 1 update per second per channel per geographic location. The database center **840*b*** updates its database so that queries can be efficiently responded to in sub-linear time.

The database centers **840*b*** can use traditional database technology. In general, the query search initiated by an audience member is not a nearest neighbor search but can be a classical textual search procedure such as a binary search. The nearest neighbor search is appropriate for the monitoring sub-system **840*a***. The database centers **840*b*** are continually updated as each new advertisement, television show or portion thereof is recognized. Standard updating algorithms can be used. However, random new entries to the database are unlikely. Rather, each new entry, or set of entries, denotes a new time segment that is later than all previously inserted items. As such, each new entry can be appended to the end of the database while still maintaining an ordered data structure that is amenable to binary and other efficient search techniques. If two entries have the same time in their time field, items can be sorted based on secondary fields such as the channel and geographic location, as depicted in FIG. 9. Since the number of such entries will be relatively small compared with the entire database, it may be sufficient to simply create a linear linked list of such entries, as depicted in FIG. 9. Of course, the size of the database is constantly increasing. As such, it may become necessary to have several levels of storage and caching. Given the envisaged application, most user queries will be for recent entries. Thus, the database may keep the last hours worth of entries in memory. If there is one entry per second for each of 100 channels in 100 geographic locations, this would correspond to 3600.times.100.times.100=36,000,000 entries which is easily accommodated in main memory. Entries that are older than one hour may be stored on disk and entries older than one week may be archived (e.g., backed up on tape) for example. The entries to this database can include time, location and channel information together with a unique identifier that is provided by the monitoring system. Of course, additional fields for each entry are also possible.

When a user query is received, the time, channel and geographic information are used to retrieve the corresponding unique identifier that is then used to access a second database that contains information associated with the identified work.

An entry 1000 in this second database is depicted in FIG. 10, which shows that associated with the unique identifier 1010, the name of a product 1020, a product category 1030, the manufacturer 1040 and the commercial's associated web site 1050. Many other data fields 1060 are also possible. Such additional fields may include fields that indicate what action should be taken on behalf of the requesting user. Example actions include simply redirecting a request to an associated Web site, or initiating an e-commerce transaction or providing an associated telephone number that may be automatically dialed if the querying device is a cell phone or displaying additional information to the user. This database is likely to be updated much less frequently, perhaps only as often as once or twice a day, as batches of new advertisements are added to the system. Alternatively, it might be updated as each new advertisement is added to the system.

An audience member (user) 810 watching a television commercial for example may react to the advertisement by initiating a query to the database center 840*b*. The device whereby the user initiates the query might be a television or set-top-box remote control, or a computer or a wireless PDA or a (WAP-enabled) cell phone or a specialized device. Typically, the query will occur during the airing of the commercial or a shortly thereafter. However, the time between the broadcasting of the advertisement and the time of the associated query is not critical and can, in some instances be much longer. For example, the audience member might bookmark the query information in a device such as a PDA or a specialized device similar to those developed by Xenote for their Itag radio linking. Later, the audience member may transmit the query to the database center 840*b*. This might happen hours or even days later.

The query contains information that the database center 840*b* uses to identify the work being viewed. This information might include the time and place where the audience member was, together with the channel being viewed. Other identifying information is also possible. The query may also contain additional information that may be used to facilitate the user's transaction and will include the return address of the user. For example, if the user is intending to order a pizza after seeing a Pizza Hut advertisement, the query may also contain personal information including his or her identity, street address and credit card information.

When the database center 840*b* receives a query, data in the query is used to identify the work and associated information. A number of possible actions are possible at this point. First, the database center 840*b* may simply function as a form of proxy server, mapping the audience member's initial query into a web address associated with the advertisement. In this case, the audience member will be sent to the corresponding Web site. The database center 840*b* may also send additional data included in the initial query to this Web site 850 in order to facilitate an e-commerce transaction between the audience member and the advertiser. In some cases, this transaction will not be direct, but may be indirect via a dealer or third party application service provider. Thus, for example, though an advertisement by Ford Motor Company may air nationally, viewers may be directed to different Web sites for Ford dealerships depending on both the audience member's and the dealerships' geographic locations. In other cases, advertisers may have contracted with the database center 840*b* to provide e-commerce capabilities. This latter arrangement has the potential to reduce the amount of traffic directed over the public Internet, restricting it, instead to a private network associated with the owner of the database center.

If the audience member (user) is not watching live television but is instead watching a taped and therefore time-shifted copy, then additional processes are needed. For the new generation of digital video recorders, irrespective of the recording media (tape or disk), it is likely to be very easy to include information identifying the location of the recorder, as well as the time and channel recorded. Location information can be provided to the recorder during the setup and installation process, for example. Digital video recorders, such as those currently manufactured by TIVO of Alviso, Calif. or Replay TV of Santa Clara, Calif. have a network connection via telephone, which can then send the query of an audience member to the database center 840*b* using the recorded rather than the current information.

In cases where query information has not been recorded, it is still possible to initiate a successful query. However, in this case, it may be necessary to extract the feature vector from the work of interest and send this information to the monitoring center 840*a* where the feature vector can be identified. This form of query is computationally more expensive but the relative number of such queries compared to those sent to the database centers 840*b* is expected to be small. It should also be noted that the physical separation of the monitoring and database centers, depicted in FIGS. 6 and 7, is not crucial to operation of the system and simply serves to more clearly separate the different functionality present in the overall system configuration.

Although the implementation architectures described above focus on the television media, it is apparent that the present invention is applicable to audio, print and other media.

§4.4 Conclusions

None of the embodiments of the invention require modification to the work or content, i.e., no active signal is embedded. Consequently, there is no change to the production processes. More importantly, from a user perspective, deployment of this system need not suffer from poor initial coverage. Provided the database is sufficiently comprehensive, early adopters will have comprehensive coverage immediately. Thus, there is less risk that the consumer will perceive that the initial performance of the deployed system is poor. Further, the present invention permits statistics to be gathered that measure users' responses to content. This information is expected to be very useful to advertisers and publishers and broadcasters.

What is claimed is:

1. A computer system comprising:

one or more electronic communications devices;

one or more processors operatively connected to the one or more electronic communications devices; and

one or more computer readable media operatively connected to the one or more processors and having stored thereon computer instructions for carrying out the steps of:

(a) maintaining, by the computer system, a database comprising:

(1) first electronic data related to identification of one or more reference electronic works; and

(2) second electronic data related to action information comprising an action to perform corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, extracted features of a first electronic work;

(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action information corresponding to the identified first electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action information with the identified first electronic work.

2. The computer system of claim 1, wherein the step of obtaining comprises receiving the first electronic work and extracting the extracted features from the first electronic work.

3. The computer system of claim 2, wherein the first electronic work is received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

4. The computer system of claim 1, wherein the step of obtaining comprises receiving the extracted features.

5. The computer system of claim 4, wherein the extracted features are received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

6. The computer system of claim 1, wherein the step of obtaining comprises receiving the first electronic work and the extracted features of the first electronic work.

7. The computer system of claim 6, wherein the first electronic work and the extracted features of the first electronic work are received from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device.

8. The computer system of claim 1, wherein at least one of the first electronic work or the extracted features of the first electronic work is obtained from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device that stores commercial transaction data, and wherein the action information comprises a commercial transaction, and the one or more computer readable media have stored thereon further computer instructions for carrying out the additional step of:

electronically performing the commercial transaction using the stored commercial transaction data.

9. The computer system of claim 1, wherein at least one of the first electronic work or the extracted features of the first electronic work is obtained from at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device, and wherein the action information comprises commercial transaction data, and the one or more computer readable media have stored thereon further computer instructions for carrying out the additional step of:

electronically performing the commercial transaction through the at least one of a set-top-box, a video recorder, a cell phone, a computer, or a portable device using the commercial transaction data.

10. The computer system of claim 8, wherein the stored commercial transaction data comprises at least one of a purchaser's name, a purchaser's address, or credit card information.

11. The computer system of claim 1, wherein the action comprises at least one of directing a user to a website related to the first electronic work, initiating an e-commerce transaction, or dialing a phone number.

12. The computer system of claim 1, wherein the action comprises providing and/or displaying additional information in association with the first electronic work.

13. A computer system comprising:

one or more electronic communications devices;

one or more processors operatively connected to the one or more electronic communications devices; and

one or more computer readable media operatively connected to the one or more processors and having stored thereon computer instructions for carrying out the steps of:

(a) maintaining, by the computer system, one or more databases comprising:

(1) first electronic data comprising a first digitally created compact electronic representation of one or more reference electronic works; and

(2) second electronic data related to an action, the action comprising providing and/or displaying an advertisement corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, a second digitally created compact electronic representation of a first electronic work;

(c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action corresponding to the matching reference electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action with the first electronic work.

14. The computer system of claim 13, wherein the action further comprises providing a link to a site on the World Wide Web associated with the advertisement.

15. The computer system of claim 13, wherein the action further comprises electronically registering a user with at least one of a service or a product related to the advertisement.

16. The computer system of claim 13, wherein the action further comprises electronically providing at least one of a coupon or a certificate related to the advertisement.

17. The computer system of claim 13, wherein the action further comprises automatically dialing a telephone number associated with the advertisement.

18. The computer system of claim 13, wherein the action further comprises collecting competitive market research data related to the advertisement.

19. The computer system of claim 13, wherein the action further comprises purchasing a product or service related to the advertisement.

20. The computer system of claim 13, wherein the action further comprises allowing a user to interact with a broadcast related to the advertisement.

21. The computer system of claim 1, further comprising the step of transmitting, by the computer system, the determined action as associated with the first electronic work.

22. The computer system of claim 1, wherein the one or more electronic works comprise at least one of a video work, an audio work, or an image work.

23. The computer system of claim 1, wherein the one or more computer readable media have stored thereon further computer instructions for carrying out the additional steps of:

(f) obtaining, by the computer system, second extracted features of a second electronic work;

(g) searching, by the computer system, for an identification of the second electronic work by comparing the second extracted features of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that the second electronic work is not identified based on results of the searching step.

24. The computer system of claim 13, wherein the one or more computer readable media have stored thereon further computer instructions for carrying out the additional steps of:

(f) obtaining, by the computer system, second extracted features of a second electronic work to be identified;

(g) searching, by the computer system, for an identification of the second electronic work by comparing the second extracted features of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that the second electronic work is not identified based on results of the searching step.

25. A computer system comprising:

one or more electronic communications devices;

one or more processors operatively connected to the one or more electronic communications devices; and

one or more computer readable media operatively connected to the one or more processors and having stored thereon computer instructions for carrying out the steps of:

(a) maintaining, by the computer system, one or more databases comprising:

(1) first electronic data comprising a first digitally created compact electronic representation comprising an extracted feature vector of one or more reference electronic works; and

(2) second electronic data related to action information, the action information comprising an action to perform corresponding to each of the one or more reference electronic works;

(b) obtaining, by the computer system, a second digitally created compact electronic representation comprising an extracted feature vector of a first electronic work;

(c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation of the first electronic work using a non-exhaustive neighbor search;

(d) determining, by the computer system, the action information corresponding to the matching reference electronic work based on the second electronic data in the database; and

(e) associating, by the computer system, the determined action information with the first electronic work.

26. The computer system of claim 25, wherein the one or more computer readable media have stored thereon further computer instructions for carrying out the additional steps of:

(f) obtaining, by the computer system, a third digitally created compact electronic representation comprising an extracted feature vector of a second electronic work;

(g) searching, by the computer system, for a matching reference electronic work that matches the second electronic work by comparing the third digitally created compact electronic representation of the second electronic work with the first electronic data in the database using a non-exhaustive neighbor search; and

(h) determining, by the computer system, that a matching reference electronic work that matches the second electronic work does not exist based on results of the searching step.

27. The computer system of claim 13, wherein the extracted feature is extracted using frequency based decomposition.

28. The computer system of claim 13, wherein the extracted feature is extracted using principal component analysis.

29. The computer system of claim 13, wherein the extracted feature is extracted using temporal sequence of feature vectors.

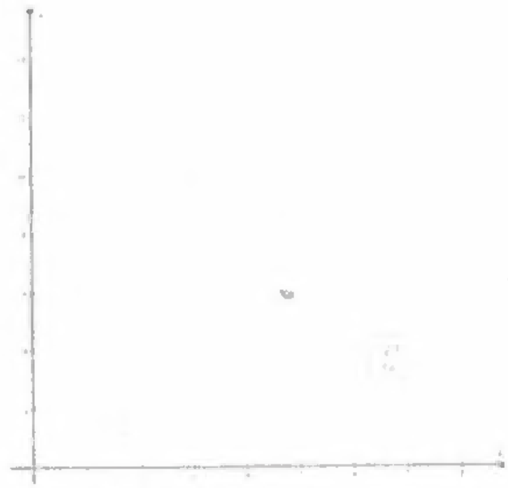
30. The computer system of claim 13, wherein the one or more computer readable media have stored thereon further computer instructions for carrying out the additional step of transmitting, by the computer system, the determined action information as associated with the first electronic identified work.

* * * * *

Big O notation

From Wikipedia, the free encyclopedia

In mathematics, **big O notation** describes the limiting behavior of a function when the argument tends towards a particular value or infinity, usually in terms of simpler functions. It is a member of a larger family of notations that is called **Landau notation**, **Bachmann–Landau notation** (after Edmund Landau and Paul Bachmann),[1][2] or **asymptotic notation**. In computer science, big O notation is used to classify algorithms by how they respond (e.g., in their processing time or working space requirements) to changes in input size.[3] In analytic number theory, it is used to estimate the "error committed" while replacing the asymptotic size, or asymptotic mean size, of an arithmetical function, by the value, or mean value, it takes at a large finite argument. A famous example is the problem of estimating the remainder term in the prime number theorem.



Example of Big O notation: $f(x) \in O(g(x))$ as there exists $c > 0$ (e.g., $c = 1$) and x_0 (e.g., $x_0 = 5$) such that $f(x) < cg(x)$ whenever $x > x_0$.

Big O notation characterizes functions according to their growth rates: different functions with the same growth rate may be represented using the same O notation. The letter O is used because the growth rate of a function is also referred to as order of the function. A description of a function in terms of big O notation usually only provides an upper bound on the growth rate of the function. Associated with big O notation are several related notations, using the symbols o, Ω, ω, and Θ, to describe other kinds of bounds on asymptotic growth rates.

Big O notation is also used in many other fields to provide similar estimates.

Contents

- 1 Formal definition
- 2 Example
- 3 Usage
  - 3.1 Infinite asymptotics
  - 3.2 Infinitesimal asymptotics
- 4 Properties
  - 4.1 Product
  - 4.2 Sum
  - 4.3 Multiplication by a constant
- 5 Multiple variables
- 6 Matters of notation
  - 6.1 Equals sign
  - 6.2 Other arithmetic operators
    - 6.2.1 Example
  - 6.3 Declaration of variables
  - 6.4 Multiple usages
- 7 Orders of common functions
- 8 Related asymptotic notations
  - 8.1 Little-o notation
  - 8.2 Big Omega notation
    - 8.2.1 The Hardy–Littlewood definition
    - 8.2.2 Simple examples
    - 8.2.3 The Knuth definition
  - 8.3 Family of Bachmann–Landau notations
  - 8.4 Use in computer science

- 8.5 Extensions to the Bachmann–Landau notations
- 9 Generalizations and related usages
- 10 History (Bachmann–Landau, Hardy, and Vinogradov notations)
- 11 See also
- 12 References and Notes
- 13 Further reading
- 14 External links

Formal definition

Let *f* and *g* be two functions defined on some subset of the real numbers. One writes

$$f(x) = O(g(x)) \text{ as } x \to \infty$$

if and only if there is a positive constant M such that for all sufficiently large values of *x*, the absolute value of *f*(*x*) is at most M multiplied by the absolute value of *g*(*x*). That is, *f*(*x*) = *O*(*g*(*x*)) if and only if there exists a positive real number *M* and a real number x_0 such that

$$|f(x)| \leq M|g(x)| \text{ for all } x \geq x_0.$$

In many contexts, the assumption that we are interested in the growth rate as the variable *x* goes to infinity is left unstated, and one writes more simply that *f*(*x*) = *O*(*g*(*x*)).

The notation can also be used to describe the behavior of *f* near some real number *a* (often, *a* = 0): we say

$$f(x) = O(g(x)) \text{ as } x \to a$$

if and only if there exist positive numbers *δ* and *M* such that

$$|f(x)| \leq M|g(x)| \text{ for } |x - a| < \delta.$$

If *g*(*x*) is non-zero for values of *x* sufficiently close to *a*, both of these definitions can be unified using the limit superior:

$$f(x) = O(g(x)) \text{ as } x \to a$$

if and only if

$$\limsup_{x \to a} \left|\frac{f(x)}{g(x)}\right| < \infty.$$

Additionally, the notation *O*(*g*(*x*)) is also used to denote the set of all functions *f*(*x*) that satisfy the relation *f*(*x*)=*O*(*g*(*x*)). In this case we write

$$f(x) \in O(g(x))$$

Example

In typical usage, the formal definition of *O* notation is not used directly; rather, the *O* notation for a function *f* is derived by the following simplification rules:

- If *f*(*x*) is a sum of several terms, the one with the largest growth rate is kept, and all others omitted.
- If *f*(*x*) is a product of several factors, any constants (terms in the product that do not depend on *x*) are omitted.

For example, let $f(x) = 6x^4 - 2x^3 + 5$, and suppose we wish to simplify this function, using *O* notation, to describe its growth rate as *x* approaches infinity. This function is the sum of three terms: $6x^4$, $-2x^3$, and 5. Of these three terms, the one with the highest growth rate is the one with the largest exponent as a function of *x*, namely $6x^4$. Now one may apply the second rule: $6x^4$ is a product of 6 and x^4 in which the first factor does not depend on x. Omitting this factor results in the simplified form x^4. Thus, we say that $f(x)$ is a "big-oh" of (x^4). Mathematically, we can write $f(x) = O(x^4)$. One may confirm this calculation using the formal definition: let $f(x) = 6x^4 - 2x^3 + 5$ and $g(x) = x^4$. Applying the formal definition from above, the statement that $f(x) = O(x^4)$ is equivalent to its expansion,

$$|f(x)| \leq M|g(x)|$$

for some suitable choice of x_0 and M and for all $x > x_0$. To prove this, let $x_0 = 1$ and $M = 13$. Then, for all $x > x_0$:

$$\begin{aligned}|6x^4 - 2x^3 + 5| &\leq 6x^4 + |2x^3| + 5 \\ &\leq 6x^4 + 2x^4 + 5x^4 \\ &= 13x^4\end{aligned}$$

so

$$|6x^4 - 2x^3 + 5| \leq 13x^4.$$

Usage

Big O notation has two main areas of application. In mathematics, it is commonly used to describe how closely a finite series approximates a given function, especially in the case of a truncated Taylor series or asymptotic expansion. In computer science, it is useful in the analysis of algorithms. In both applications, the function $g(x)$ appearing within the $O(\ldots)$ is typically chosen to be as simple as possible, omitting constant factors and lower order terms. There are two formally close, but noticeably different, usages of this notation: infinite asymptotics and infinitesimal asymptotics. This distinction is only in application and not in principle, however—the formal definition for the "big O" is the same for both cases, only with different limits for the function argument.

Infinite asymptotics

Big O notation is useful when analyzing algorithms for efficiency. For example, the time (or the number of steps) it takes to complete a problem of size n might be found to be $T(n) = 4n^2 - 2n + 2$. As n grows large, the n^2 term will come to dominate, so that all other terms can be neglected — for instance when $n = 500$, the term $4n^2$ is 1000 times as large as the $2n$ term. Ignoring the latter would have negligible effect on the expression's value for most purposes. Further, the coefficients become irrelevant if we compare to any other order of expression, such as an expression containing a term n^3 or n^4. Even if $T(n) = 1{,}000{,}000n^2$, if $U(n) = n^3$, the latter will always exceed the former once n grows larger than 1,000,000 ($T(1{,}000{,}000) = 1{,}000{,}000^3 = U(1{,}000{,}000)$). Additionally, the number of steps depends on the details of the machine model on which the algorithm runs, but different types of machines typically vary by only a constant factor in the number of steps needed to execute an algorithm. So the big O notation captures what remains: we write either

$$T(n) = O(n^2)$$

or

$$T(n) \in O(n^2)$$

and say that the algorithm has *order of n^2* time complexity. Note that "=" is not meant to express "is equal to" in its normal mathematical sense, but rather a more colloquial "is", so the second expression is technically accurate (see the "Equals sign" discussion below) while the first is considered by some as an abuse of notation.[3]

Infinitesimal asymptotics

Big O can also be used to describe the error term in an approximation to a mathematical function. The most significant terms are written explicitly, and then the least-significant terms are summarized in a single big O term. Consider, for example, the exponential series and two expressions of it that are valid when x is small:

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots && \text{for all } x \\ &= 1 + x + \frac{x^2}{2} + O(x^3) && \text{as } x \to 0, \\ &= 1 + x + O(x^2) && \text{as } x \to 0. \end{aligned}$$

The second expression (the one with $O(x^3)$) means the absolute-value of the error $e^x - (1 + x + x^2/2)$ is smaller than some constant times $|x^3|$ when x is close enough to 0.

Properties

If the function f can be written as a finite sum of other functions, then the fastest growing one determines the order of $f(n)$. For example

$$f(n) = 9\log n + 5(\log n)^3 + 3n^2 + 2n^3 = O(n^3), \qquad \text{as } n \to \infty.$$

In particular, if a function may be bounded by a polynomial in n, then as n tends to *infinity*, one may disregard *lower-order* terms of the polynomial. Another thing to notice is the sets $O(n^c)$ and $O(c^n)$ are very different. If c is greater than one, then the latter grows much faster. A function that grows faster than n^c for any c is called *superpolynomial*. One that grows more slowly than any exponential function of the form c^n is called *subexponential*. An algorithm can require time that is both superpolynomial and subexponential; examples of this include the fastest known algorithms for integer factorization and the function $n^{\log n}$.

We may ignore any powers of n inside of the logarithms. The set $O(\log n)$ is exactly the same as $O(\log(n^c))$. The logarithms differ only by a constant factor (since $\log(n^c) = c \log n$) and thus the big O notation ignores that. Similarly, logs with different constant bases are equivalent. On the other hand, exponentials with different bases are not of the same order. For example, 2^n and 3^n are not of the same order.

Changing units may or may not affect the order of the resulting algorithm. Changing units is equivalent to multiplying the appropriate variable by a constant wherever it appears. For example, if an algorithm runs in the order of n^2, replacing n by cn means the algorithm runs in the order of c^2n^2, and the big O notation ignores the constant c^2. This can be written as $c^2n^2 = O(n^2)$. If, however, an algorithm runs in the order of 2^n, replacing n with cn gives $2^{cn} = (2^c)^n$. This is not equivalent to 2^n in general. Changing variables may also affect the order of the resulting algorithm. For example, if an algorithm's run time is $O(n)$ when measured in terms of the number n of *digits* of an input number x, then its run time is $O(\log x)$ when measured as a function of the input number x itself, because $n = O(\log x)$.

Product

$$f_1 = O(g_1) \text{ and } f_2 = O(g_2) \Rightarrow f_1 f_2 \in O(g_1 g_2)$$
$$f \cdot O(g) = O(fg)$$

Sum

$$f_1 = O(g_1) \text{ and } f_2 = O(g_2) \Rightarrow f_1 + f_2 = O(|g_1| + |g_2|)$$

This implies $f_1 = O(g)$ and $f_2 = O(g) \Rightarrow f_1 + f_2 \in O(g)$, which means that $O(g)$ is a convex cone.

If f and g are positive functions, $f + O(g) = O(f + g)$

Multiplication by a constant

Let k be a constant. Then:
$O(kg) = O(g)$ if k is nonzero.
$f \in O(g) \Rightarrow kf = O(g)$.

Multiple variables

Big O (and little o, and Ω...) can also be used with multiple variables. To define Big O formally for multiple variables, suppose $f(\vec{x})$

and $g(\vec{x})$ are two functions defined on some subset of $\mathbb{R}^n$. We say

$$f(\vec{x}) \text{ is } O(g(\vec{x})) \text{ as } \vec{x} \to \infty$$

if and only if[5]

$$\exists M\ \exists C > 0 \text{ such that for all } \vec{x} \text{ with } x_i \geq M \text{ for some } i, |f(\vec{x})| \leq C|g(\vec{x})|.$$

Equivalently, the condition that $x_i \geq M$ for some i can be replaced with the condition that $\|\vec{x}\|_\infty \geq M$, where $\|\vec{x}\|_\infty$ denotes the Chebyshev distance. For example, the statement

$$f(n,m) = n^2 + m^3 + O(n+m) \text{ as } n,m \to \infty$$

asserts that there exist constants C and M such that

$$\forall \|(n,m)\|_\infty \geq M : |g(n,m)| \leq C(n+m),$$

where $g(n,m)$ is defined by

$$f(n,m) = n^2 + m^3 + g(n,m).$$

Note that this definition allows all of the coordinates of $\vec{x}$ to increase to infinity. In particular, the statement

$$f(n,m) = O(n^m) \text{ as } n,m \to \infty$$

(i.e., $\exists C\ \exists M\ \forall n\ \forall m \ldots$) is quite different from

$$\forall m : f(n,m) = O(n^m) \text{ as } n \to \infty$$

(i.e., $\forall m\ \exists C\ \exists M\ \forall n \ldots$).

This is not the only generalization of big O to multivariate functions, and in practice, there is some inconsistency in the choice of definition.[6]

Matters of notation

Equals sign

The statement "$f(x)$ is $O(g(x))$" as defined above is usually written as $f(x) = O(g(x))$. Some consider this to be an abuse of notation, since the use of the equals sign could be misleading as it suggests a symmetry that this statement does not have. As de Bruijn says, $O(x) = O(x^2)$ is true but $O(x^2) = O(x)$ is not.[7] Knuth describes such statements as "one-way equalities", since if the sides could be reversed, "we could deduce ridiculous things like $n = n^2$ from the identities $n = O(n^2)$ and $n^2 = O(n^2)$."[8] For these reasons, it would be more precise to use set notation and write $f(x) \in O(g(x))$, thinking of $O(g(x))$ as the class of all functions $h(x)$ such that $|h(x)| \leq C|g(x)|$ for some constant C.[8] However, the use of the equals sign is customary. Knuth pointed out that "mathematicians customarily use the = sign as they use the word 'is' in English: Aristotle is a man, but a man isn't necessarily Aristotle."[9]

Other arithmetic operators

Big O notation can also be used in conjunction with other arithmetic operators in more complicated equations. For example, $h(x) + O(f(x))$ denotes the collection of functions having the growth of $h(x)$ plus a part whose growth is limited to that of $f(x)$. Thus,

$$g(x) = h(x) + O(f(x))$$

expresses the same as

$$g(x) - h(x) = O(f(x)).$$

Example

Suppose an algorithm is being developed to operate on a set of n elements. Its developers are interested in finding a function $T(n)$ that will express how long the algorithm will take to run (in some arbitrary measurement of time) in terms of the number of elements in the input set. The algorithm works by first calling a subroutine to sort the elements in the set and then perform its own operations. The sort has a known time complexity of $O(n^2)$, and after the subroutine runs the algorithm must take an additional $55n^3 + 2n + 10$ steps before it terminates. Thus the overall time complexity of the algorithm can be expressed as $T(n) = 55n^3 + O(n^2)$. Here the terms $2n+10$ are subsumed within the faster-growing $O(n^2)$. Again, this usage disregards some of the formal meaning of the "=" symbol, but it does allow one to use the big O notation as a kind of convenient placeholder.

Declaration of variables

Another feature of the notation, although less exceptional, is that function arguments may need to be inferred from the context when several variables are involved. The following two right-hand side big O notations have dramatically different meanings:

$$f(m) = O(m^n)\,,$$
$$g(n)\ = O(m^n)\,.$$

The first case states that $f(m)$ exhibits polynomial growth, while the second, assuming $m > 1$, states that $g(n)$ exhibits exponential growth. To avoid confusion, some authors use the notation

$$g(x) = O(f(x))\,.$$

rather than the less explicit

$$g = O(f)\,.$$

Multiple usages

In more complicated usage, $O(...)$ can appear in different places in an equation, even several times on each side. For example, the following are true for $n \to \infty$

$$(n+1)^2 = n^2 + O(n)$$
$$(n + O(n^{1/2}))(n + O(\log n))^2 = n^3 + O(n^{5/2})$$
$$n^{O(1)} = O(e^n).$$

The meaning of such statements is as follows: for *any* functions which satisfy each $O(...)$ on the left side, there are *some* functions satisfying each $O(...)$ on the right side, such that substituting all these functions into the equation makes the two sides equal. For example, the third equation above means: "For any function $f(n) = O(1)$, there is some function $g(n) = O(e^n)$ such that $n^{f(n)} = g(n)$." In terms of the "set notation" above, the meaning is that the class of functions represented by the left side is a subset of the class of functions represented by the right side. In this use the "=" is a formal symbol that unlike the usual use of "=" is not a symmetric relation. Thus for example $n^{O(1)} = O(e^n)$ does not imply the false statement $O(e^n) = n^{O(1)}$

Orders of common functions

Further information: Time complexity § Table of common time complexities

Here is a list of classes of functions that are commonly encountered when analyzing the running time of an algorithm. In each case, c is a constant and n increases without bound. The slower-growing functions are generally listed first.

Notation	Name	Example
$O(1)$	constant	Determining if a binary number is even or odd; Calculating $(-1)^n$; Using a constant-size lookup table
$O(\log\log n)$	double logarithmic	Number of comparisons spent finding an item using interpolation search in a sorted array of uniformly distributed values
$O(\log n)$	logarithmic	Finding an item in a sorted array with a binary search or a balanced search tree as well as all operations in a Binomial heap
$O((\log n)^c),\ c > 1$	polylogarithmic	Matrix chain ordering can be solved in polylogarithmic time on a Parallel Random Access Machine.
$O(n^c),\ 0 < c < 1$	fractional power	Searching in a kd-tree
$O(n)$	linear	Finding an item in an unsorted list or a malformed tree (worst case) or in an unsorted array; adding two n-bit integers by ripple carry
$O(n\log^* n)$	n log-star n	Performing triangulation of a simple polygon using Seidel's algorithm, or the union–find algorithm. Note that $\log^*(n) = \begin{cases} 0, & \text{if } n \le 1 \\ 1 + \log^*(\log n), & \text{if } n > 1 \end{cases}$
$O(n\log n) = O(\log n!)$	linearithmic, loglinear, or quasilinear	Performing a fast Fourier transform; heapsort, quicksort (best and average case), or merge sort
$O(n^2)$	quadratic	Multiplying two n-digit numbers by a simple algorithm; bubble sort (worst case or naive implementation), Shell sort, quicksort (worst case), selection sort or insertion sort
$O(n^c),\ c > 1$	polynomial or algebraic	Tree-adjoining grammar parsing; maximum matching for bipartite graphs
$L_n[\alpha, c],\ 0 < \alpha < 1 = e^{(c+o(1))(\ln n)^\alpha(\ln\ln n)^{1-\alpha}}$	L-notation or sub-exponential	Factoring a number using the quadratic sieve or number field sieve
$O(c^n),\ c > 1$	exponential	Finding the (exact) solution to the travelling salesman problem using dynamic programming; determining if two logical statements are equivalent using brute-force search
$O(n!)$	factorial	Solving the traveling salesman problem via brute-force search; generating all unrestricted permutations of a poset; finding the determinant with expansion by minors; enumerating all partitions of a set

The statement $f(n) = O(n!)$ is sometimes weakened to $f(n) = O\left(n^n\right)$ to derive simpler formulas for asymptotic complexity. For any $k > 0$ and $c > 0$, $O(n^c(\log n)^k)$ is a subset of $O(n^{c+\varepsilon})$ for any $\varepsilon > 0$, so may be considered as a polynomial with some bigger order.

Related asymptotic notations

Big *O* is the most commonly used asymptotic notation for comparing functions, although in many cases Big *O* may be replaced with Big Theta Θ for asymptotically tighter bounds. Here, we define some related notations in terms of Big *O*, progressing up to the family of Bachmann–Landau notations to which Big *O* notation belongs.

Little-o notation

"Little o" redirects here. For the baseball player, see Omar Vizquel.

The informal assertion "$f(x)$ is little-o of $g(x)$" is formally written $f(x) = o(g(x))$, or in set notation $f(x) \in o(g(x))$. Intuitively, it means that $g(x)$ grows much faster than $f(x)$, or similarly, that the growth of $f(x)$ is nothing compared to that of $g(x)$. It assumes that f and g are both functions of one variable. Formally, $f(n) = o(g(n))$ (or $f(n) \in o(g(n))$) as $n \to \infty$ means that for every positive constant ε there exists a constant N such that

$$|f(n)| \le \epsilon|g(n)| \qquad \text{for all } n \ge N.$$ [9]

Note the difference between the earlier formal definition for the big-O notation, and the present definition of little-o: while the former has to be true for *at least one* constant *M* the latter must hold for *every* positive constant ε, however small.[9] In this way little-o notation makes a stronger statement than the corresponding big-O notation: every function that is little-o of *g* is also big-O of *g*, but not every function that is big-O of *g* is also little-o of *g* (for instance *g* itself is not, unless it is identically zero near ∞).

If *g*(*x*) is nonzero, or at least becomes nonzero beyond a certain point, the relation *f*(*x*) = *o*(*g*(*x*)) is equivalent to

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0.$$

For example,

- $2x = o(x^2)$
- $2x^2 \neq o(x^2)$
- $1/x = o(1)$

Little-o notation is common in mathematics but rarer in computer science. In computer science the variable (and function value) is most often a natural number. In mathematics, the variable and function values are often real numbers. The following properties (expressed in the more recent, set-theoretical notation) can be useful:

- $c \cdot o(f) = o(f)$ for $c \neq 0$
- $o(f)o(g) \subseteq o(fg)$
- $o(o(f)) \subseteq o(f)$
- $o(f) \subset O(f)$ (and thus the above properties apply with most combinations of o and O).

As with big O notation, the statement "$f(x)$ is $o(g(x))$" is usually written as $f(x) = o(g(x))$, which some consider an abuse of notation.

Big Omega notation

There are two very widespread and incompatible definitions of the statement

$$f(x) = \Omega(g(x)) \ (x \to a),$$

where *a* is some real number, ∞, or −∞, where *f* and *g* are real functions defined in a neighbourhood of *a*, and where *g* is positive in this neighbourhood.

The first one (chronologically) is used in analytic number theory, and the other one in computational complexity theory. When the two subjects meet, this situation is bound to generate confusion.

The Hardy–Littlewood definition

In 1914 G.H. Hardy and J.E. Littlewood introduced the new symbol Ω,[10] which is defined as follows:

$$f(x) = \Omega(g(x)) \ (x \to \infty) \ \Leftrightarrow \ \limsup_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| > 0$$

Thus $f(x) = \Omega(g(x))$ is the negation of $f(x) = o(g(x))$.

In 1918 the same authors introduced the two new symbols Ω_R and Ω_L,[11] thus defined:

$$f(x) = \Omega_R(g(x)) \ (x \to \infty) \ \Leftrightarrow \ \limsup_{x \to \infty} \frac{f(x)}{g(x)} > 0$$

$$f(x) = \Omega_L(g(x)) \ (x \to \infty) \ \Leftrightarrow \ \liminf_{x \to \infty} \frac{f(x)}{g(x)} < 0$$

Hence $f(x) = \Omega_R(g(x))$ is the negation of $f(x) < o(g(x))$, and $f(x) = \Omega_L(g(x))$ the negation of $f(x) > o(g(x))$.

Contrary to a later assertion of D.E. Knuth,[12] Edmund Landau did use these three symbols, with the same meanings, in 1924.[13]

These Hardy-Littlewood symbols are prototypes, which after Landau were never used again exactly thus.

Ω_R became Ω_+, and Ω_L became Ω_-.

These three symbols $\Omega, \Omega_+, \Omega_-$, as well as $f(x) = \Omega_\pm(g(x))$ (meaning that $f(x) = \Omega_+(g(x))$ and $f(x) = \Omega_-(g(x))$ are both satisfied), are now currently used in analytic number theory.[14]

Simple examples

We have

$$\sin x = \Omega(1)\ (x \to \infty),$$

and more precisely

$$\sin x = \Omega_\pm(1)\ (x \to \infty).$$

We have

$$\sin x + 1 = \Omega(1)\ (x \to \infty),$$

and more precisely

$$\sin x + 1 = \Omega_+(1)\ (x \to \infty);$$

however

$$\sin x + 1 \neq \Omega_-(1)\ (x \to \infty).$$

The Knuth definition

In 1976 D.E. Knuth published a paper to justify his use of the Ω-symbol to describe a stronger property. Knuth wrote: "For all the applications I have seen so far in computer science, a stronger requirement [...] is much more appropriate". He defined

$$f(x) = \Omega(g(x)) \Leftrightarrow g(x) = O(f(x))$$

with the comment: "Although I have changed Hardy and Littlewood's definition of Ω, I feel justified in doing so because their definition is by no means in wide use, and because there are other ways to say what they want to say in the comparatively rare cases when their definition applies".[12] However, the Hardy–Littlewood definition had been used for at least 25 years.[15]

Family of Bachmann–Landau notations

Notation	Name	Intuition	Informal definition: for sufficiently large n...	Formal Definition
$f(n) = O(g(n))$	Big O; Big Oh; Big Omicron[12]	f is bounded above by g (up to constant factor) asymptotically	$\|f(n)\| \le k \cdot \|g(n)\|$ for some positive k	$\exists k > 0\ \exists n_0\ \forall n > n_0\ \|f(n)\| \le k \cdot \|g(n)\|$
$f(n) = \Omega(g(n))$	Big Omega	**Two definitions :** Number theory: f is not dominated by g asymptotically. Complexity theory: f is bounded below by g asymptotically	Number theory: $f(n) \ge k \cdot g(n)$ for infinitely many values of n and for some positive k. Complexity theory: $f(n) \ge k \cdot g(n)$ for some positive k	Number theory: $\exists k > 0\ \forall n_0\ \exists n > n_0\ f(n) \ge k \cdot g(n)$. Complexity theory: $\exists k > 0\ \exists n_0\ \forall n > n_0\ f(n) \ge k \cdot g(n)$
$f(n) = \Theta(g(n))$	Big Theta	f is bounded both above and below by g asymptotically	$k_1 \cdot g(n) \le f(n) \le k_2 \cdot g(n)$ for some positive k_1, k_2	$\exists k_1 > 0\ \exists k_2 > 0\ \exists n_0\ \forall n > n_0$ $k_1 \cdot g(n) \le f(n) \le k_2 \cdot g(n)$
$f(n) = o(g(n))$	Small O; Small Oh	f is dominated by g asymptotically	$\|f(n)\| \le k \cdot \|g(n)\|$, for every fixed positive number k	$\forall k > 0\ \exists n_0\ \forall n > n_0\ \|f(n)\| \le k \cdot \|g(n)\|$
$f(n) = \omega(g(n))$	Small Omega	f dominates g asymptotically	$\|f(n)\| \ge k \cdot \|g(n)\|$, for every fixed positive number k	$\forall k > 0\ \exists n_0\ \forall n > n_0\ \|f(n)\| \ge k \cdot \|g(n)\|$
$f(n) \sim g(n)$	On the order of	f is equal to g asymptotically	$f(n)/g(n) \to 1$	$\forall \varepsilon > 0\ \exists n_0\ \forall n > n_0\ \left\| \frac{f(n)}{g(n)} - 1 \right\| < \varepsilon$

Aside from the Big O notation, the Big Theta Θ and Big Omega Ω notations are the two most often used in computer science; the small omega ω notation is occasionally used in computer science.

Aside from the Big O notation, the small o, Big Omega Ω and ∼ notations are the three most often used in number theory; the small omega ω notation is never used in number theory.

Use in computer science

For more details on this topic, see Analysis of algorithms.

Informally, especially in computer science, the Big O notation often is permitted to be somewhat abused to describe an asymptotic tight bound where using Big Theta Θ notation might be more factually appropriate in a given context. For example, when considering a function $T(n) = 73n^3 + 22n^2 + 58$, all of the following are generally acceptable, but tighter bounds (i.e., numbers 2 and 3 below) are usually strongly preferred over looser bounds (i.e., number 1 below).

1. $T(n) = O(n^{100})$

2. $T(n) = O(n^3)$
3. $T(n) = \Theta(n^3)$

The equivalent English statements are respectively:

1. $T(n)$ grows asymptotically no faster than n^{100}
2. $T(n)$ grows asymptotically no faster than n^3
3. $T(n)$ grows asymptotically as fast as n^3

So while all three statements are true, progressively more information is contained in each. In some fields, however, the Big O notation (number 2 in the lists above) would be used more commonly than the Big Theta notation (bullets number 3 in the lists above). For example, if $T(n)$ represents the running time of a newly developed algorithm for input size n, the inventors and users of the algorithm might be more inclined to put an upper asymptotic bound on how long it will take to run without making an explicit statement about the lower asymptotic bound.

Extensions to the Bachmann–Landau notations

Another notation sometimes used in computer science is $\tilde{O}$ (read *soft-O*): $f(n) = \tilde{O}(g(n))$ is shorthand for $f(n) = O(g(n) \log^k g(n))$ for some k. Essentially, it is Big O notation, ignoring logarithmic factors because the growth-rate effects of some other super-logarithmic function indicate a growth-rate explosion for large-sized input parameters that is more important to predicting bad run-time performance than the finer-point effects contributed by the logarithmic-growth factor(s). This notation is often used to obviate the "nitpicking" within growth-rates that are stated as too tightly bounded for the matters at hand (since $\log^k n$ is always $o(n^\varepsilon)$ for any constant k and any $\varepsilon > 0$).

Also the L notation, defined as

$$L_n[\alpha, c] = O\left(e^{(c+o(1))(\ln n)^\alpha(\ln \ln n)^{1-\alpha}}\right),$$

is convenient for functions that are between polynomial and exponential.

Generalizations and related usages

The generalization to functions taking values in any normed vector space is straightforward (replacing absolute values by norms), where f and g need not take their values in the same space. A generalization to functions g taking values in any topological group is also possible. The "limiting process" $x \to x_o$ can also be generalized by introducing an arbitrary filter base, i.e. to directed nets f and g. The o notation can be used to define derivatives and differentiability in quite general spaces, and also (asymptotical) equivalence of functions,

$$f \sim g \iff (f - g) \in o(g)$$

which is an equivalence relation and a more restrictive notion than the relationship "f is $\Theta(g)$" from above. (It reduces to $\lim f / g = 1$ if f and g are positive real valued functions.) For example, $2x$ is $\Theta(x)$, but $2x - x$ is not $o(x)$.

History (Bachmann–Landau, Hardy, and Vinogradov notations)

The symbol O was first introduced by number theorist Paul Bachmann in 1894, in the second volume of his book *Analytische Zahlentheorie* ("analytic number theory"), the first volume of which (not yet containing big O notation) was published in 1892.[16] The number theorist Edmund Landau adopted it, and was thus inspired to introduce in 1909 the notation o;[17] hence both are now called Landau symbols. These notations were used in applied mathematics during the 1950s for asymptotic analysis.[18] The big O was popularized in computer science by Donald Knuth, who re-introduced the related Omega and Theta notations.[12] Knuth also noted that the Omega notation had been introduced by Hardy and Littlewood[10] under a different meaning "≠o" (i.e. "is not an o of"), and proposed the above definition. Hardy and Littlewood's original definition (which was also used in one paper by Landau[13]) is still used in number theory (where Knuth's definition is never used). In fact, Landau also used in 1924, in the paper just mentioned, the symbols Ω_R ("right") and Ω_L ("left"), which were introduced in 1918 by Hardy and Littlewood,[11] and which were precursors for the modern symbols Ω_+ ("is not smaller than a small o of") and Ω_- ("is not larger than a small o of"). Thus the Omega symbols (with their original meanings) are sometimes also referred to as "Landau symbols".

Also, Landau never used the Big Theta and small omega symbols.

Hardy's symbols were (in terms of the modern O notation)

$$f \preceq g \iff f \in O(g) \text{ and } f \prec g \iff f \in o(g);$$

(Hardy however never defined or used the notation $\prec\!\!\prec$, nor $\ll$, as it has been sometimes reported). It should also be noted that Hardy introduces the symbols $\preceq$ and $\prec$ (as well as some other symbols) in his 1910 tract "Orders of Infinity", and makes use of it only in three papers (1910–1913). In his nearly 400 remaining papers and books he consistently uses the Landau symbols O and o.

Hardy's notation is not used anymore. On the other hand, in the 1930s,[19] the Russian number theorist Ivan Matveyevich Vinogradov introduced his notation $\ll$, which has been increasingly used in number theory instead of the O notation. We have

$$f \ll g \iff f \in O(g),$$

and frequently both notations are used in the same paper.

The big-O originally stands for "order of" ("Ordnung", Bachmann 1894), and is thus a roman letter. Neither Bachmann nor Landau ever call it "Omicron". The symbol was much later on (1976) viewed by Knuth as a capital omicron,[12] probably in reference to his definition of the symbol Omega. The digit zero should not be used.

See also

- Asymptotic expansion: Approximation of functions generalizing Taylor's formula
- Asymptotically optimal: A phrase frequently used to describe an algorithm that has an upper bound asymptotically within a constant of a lower bound for the problem
- Big O in probability notation: O_p,o_p
- Limit superior and limit inferior: An explanation of some of the limit notation used in this article
- Nachbin's theorem: A precise method of bounding complex analytic functions so that the domain of convergence of integral transforms can be stated

References and Notes

1. Homayoon Beigi (9 December 2011). *Fundamentals of Speaker Recognition* (http://books.google.com/books?id=qIMDvu3gJCQC&pg=PA777). Springer. p. 777. ISBN 978-0-387-77592-0.
2. Mark H. Holmes (5 December 2012). *Introduction to Perturbation Methods* (http://books.google.com/books?id=EX5iNgIbqB4C&pg=PA4). Springer. pp. 4–. ISBN 978-1-4614-5477-9.
3. Mohr, Austin. "Quantum Computing in Complexity Theory and Theory of Computation" (http://www.austinmohr.com/Work_files/complexity.pdf) (PDF). p. 2. Retrieved 7 June 2014.
4. Thomas H. Cormen et al., 2001, Introduction to Algorithms, Second Edition (http://highered.mcgraw-hill.com/sites/0070131511/)
5. Cormen, Thomas; Leiserson, Charles; Rivest, Ronald; Stein, Clifford (2009). *Introduction to Algorithms* (Third ed.). MIT. p. 53.
6. Howell, Rodney. "On Asymptotic Notation with Multiple Variables" (http://people.cis.ksu.edu/~rhowell/asymptotic.pdf) (PDF). Retrieved 2015-04-23.
7. N. G. de Bruijn (1958). *Asymptotic Methods in Analysis* (http://books.google.com/?id=_tnwmvHmVwMC&pg=PA5&vq=%22The+trouble+is%22). Amsterdam: North-Holland. pp. 5–7. ISBN 978-0-486-64221-5.
8. Ronald Graham, Donald Knuth, and Oren Patashnik (1994). 0-201-55802-5 *Concrete Mathematics* (http://books.google.com/?id=pntQAAAAMAAJ&dq=editions:ISBN) (2 ed.). Reading, Massachusetts: Addison–Wesley. p. 446. ISBN 978-0-201-55802-9.
9. Donald Knuth (June–July 1998). "Teach Calculus with Big O" (http://www.ams.org/notices/199806/commentary.pdf) (PDF). *Notices of the American Mathematical Society* **45** (6): 687. (Unabridged version (http://www-cs-staff.stanford.edu/~knuth/ocalc.tex))
10. G. H. Hardy and J. E. Littlewood, "Some problems of Diophantine approximation", Acta Mathematica 37 (1914), p. 225
11. G. H. Hardy and J. E. Littlewood, « Contribution to the theory of the Riemann zeta-function and the theory of the distribution of primes », *Acta Mathematica*, vol. 41, 1918
12. Donald Knuth. "Big Omicron and big Omega and big Theta (http://www.phil.uu.nl/datastructuren/10-11/knuth_big_omicron.pdf)", SIGACT News, Apr.-June 1976, 18-24.
13. E. Landau, "Über die Anzahl der Gitterpunkte in gewissen Bereichen. IV." Nachr. Gesell. Wiss. Gött. Math-phys. Kl. 1924, 137–150.
14. Aleksandar Ivić. The Riemann zeta-function, chapter 9. John Wiley & Sons 1985.
15. E. C. Titchmarsh, The Theory of the Riemann Zeta-Function (Oxford; Clarendon Press, 1951)
16. Nicholas J. Higham, *Handbook of writing for the mathematical sciences*, SIAM. ISBN 0-89871-420-6, p. 25
17. Edmund Landau. Handbuch der Lehre von der Verteilung der Primzahlen, Teubner, Leipzig 1909, p.883.
18. Erdelyi, A. (1956). *Asymptotic Expansions*. ISBN 978-0486603186.
19. See for instance "A new estimate for *G(n)* in Waring's problem" (Russian). Doklady Akademii Nauk SSSR 5, No 5-6 (1934), 249-253. Translated in English in: Selected works / Ivan Matveevič Vinogradov ; prepared by the Steklov Mathematical Institute of the Academy of Sciences of the USSR on the occasion of his 90th birthday. Springer-Verlag, 1985.

Further reading

- Paul Bachmann. *Die Analytische Zahlentheorie. Zahlentheorie*. pt. 2 Leipzig: B. G. Teubner, 1894.
- Edmund Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. 2 vols. Leipzig: B. G. Teubner, 1909.
- G. H. Hardy. *Orders of Infinity: The 'Infinitärcalcül' of Paul du Bois-Reymond*, 1910.
- Donald Knuth. *The Art of Computer Programming*, Volume 1: *Fundamental Algorithms*, Third Edition. Addison–Wesley, 1997. ISBN 0-201-89683-4. Section 1.2.11: Asymptotic Representations, pp. 107–123.
- Paul Vitanyi, Lambert Meertens, Big Omega versus the wild functions, Bull. European Association Theoret. Comput. Sci. (EATCS) 22(1984), 14-19. Also: ACM-SIGACT News, 16:4(1984) 56-59.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw–Hill, 2001. ISBN 0-262-03293-7. Section 3.1: Asymptotic notation, pp. 41–50.
- Michael Sipser (1997). *Introduction to the Theory of Computation*. PWS Publishing. ISBN 0-534-94728-X. Pages 226–228 of section 7.1: Measuring complexity.
- Jeremy Avigad, Kevin Donnelly. *Formalizing O notation in Isabelle/HOL (http://www.andrew.cmu.edu/~avigad/Papers/bigo.pdf)*
- Paul E. Black, "big-O notation" (http://www.nist.gov/dads/HTML/bigOnotation.html), in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 11 March 2005. Retrieved December 16, 2006.
- Paul E. Black, "little-o notation" (http://www.nist.gov/dads/HTML/littleOnotation.html), in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 17 December 2004. Retrieved December 16, 2006.
- Paul E. Black, "Ω" (http://www.nist.gov/dads/HTML/omegaCapital.html), in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 17 December 2004. Retrieved December 16, 2006.
- Paul E. Black, "ω" (http://www.nist.gov/dads/HTML/omega.html), in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 29 November 2004. Retrieved December 16, 2006.
- Paul E. Black, "Θ" (http://www.nist.gov/dads/HTML/theta.html), in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 17 December 2004. Retrieved December 16, 2006.

External links

- Growth of sequences — OEIS (Online Encyclopedia of Integer Sequences) Wiki (http://oeis.org/wiki/Growth_of_sequences)
- Introduction to Asymptotic Notations (http://www.soe.ucsc.edu/classes/cmps102/Spring04/TantaloAsymp.pdf)
- Landau Symbols (http://mathworld.wolfram.com/LandauSymbols.html)
- Big-O Notation – What is it good for (http://www.perlmonks.org/?node_id=573138)

The Wikibook *Data Structures* has a page on the topic of: ***Big-O Notation***

Retrieved from "https://en.wikipedia.org/w/index.php?title=Big_O_notation&oldid=678755202"

Categories: Mathematical notation | Asymptotic analysis | Analysis of algorithms

- This page was last modified on 31 August 2015, at 12:35.

US006188010B1

(12) **United States Patent**
Iwamura

(10) **Patent No.: US 6,188,010 B1**
(45) **Date of Patent: Feb. 13, 2001**

(54) **MUSIC SEARCH BY MELODY INPUT**

(75) Inventor: **Ryuichi Iwamura**, San Diego, CA (US)

(73) Assignees: **Sony Corporation**, Tokyo (JP); **Sony Electronics, Inc.**, Park Ridge, NJ (US)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **09/429,260**

(22) Filed: **Oct. 29, 1999**

(51) **Int. Cl.**[7] **G09B 15/04**; G10H 1/26
(52) **U.S. Cl.** .... **84/609**; 84/477 R
(58) **Field of Search** .... 84/609–614, 634–638, 84/477 R, 478; 434/307 A

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,402,339		3/1995	Nakashima et al. .	
5,670,730	*	9/1997	Grewe et al.	84/609
5,739,451		4/1998	Winsky et al. .	
5,864,868	*	1/1999	Contois	84/609 X
5,895,876	*	4/1999	Moriyama	84/609
5,925,843	*	7/1999	Miller et al.	84/609
5,969,283	*	10/1999	Looney et al.	84/609
5,986,200	*	11/1999	Curtin	84/609
6,031,174	*	2/2000	Takabayashi	84/609

OTHER PUBLICATIONS

MIDISOFT Desktop Sheet Music 2.0 and MIDISOFT Studio 6.0 http://www.midisoft.com/upgrade.htm Apr. 10, 1999.

* cited by examiner

Primary Examiner—Stanley J. Witkowski
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

A method to enable one to search for a song title when only its melody is known. An interface is used which allows the user to enter the melody in an easy to understand manner. A remote music database with melody information is searched for the melody entered by the user, using, for example, a peak or differential matching algorithm. Upon receiving a search request, a database server, i.e., a remote computer, which can be accessed by the user via a client PC, sends a web page, for example, in HTML or Java containing the search results to the client PC. The client PC receives and displays the search results on the monitor and exchanges data with the server. Sound may also be played through the sound card to connected speakers.

40 Claims, 8 Drawing Sheets

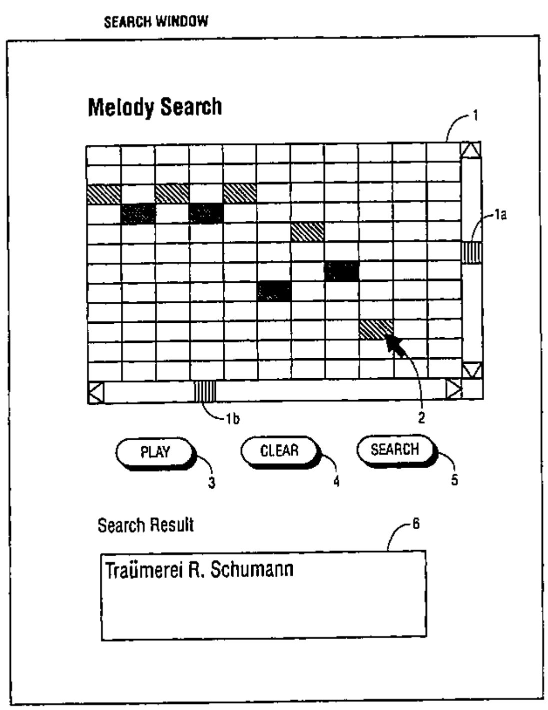


FIG. 1 SEARCH WINDOW

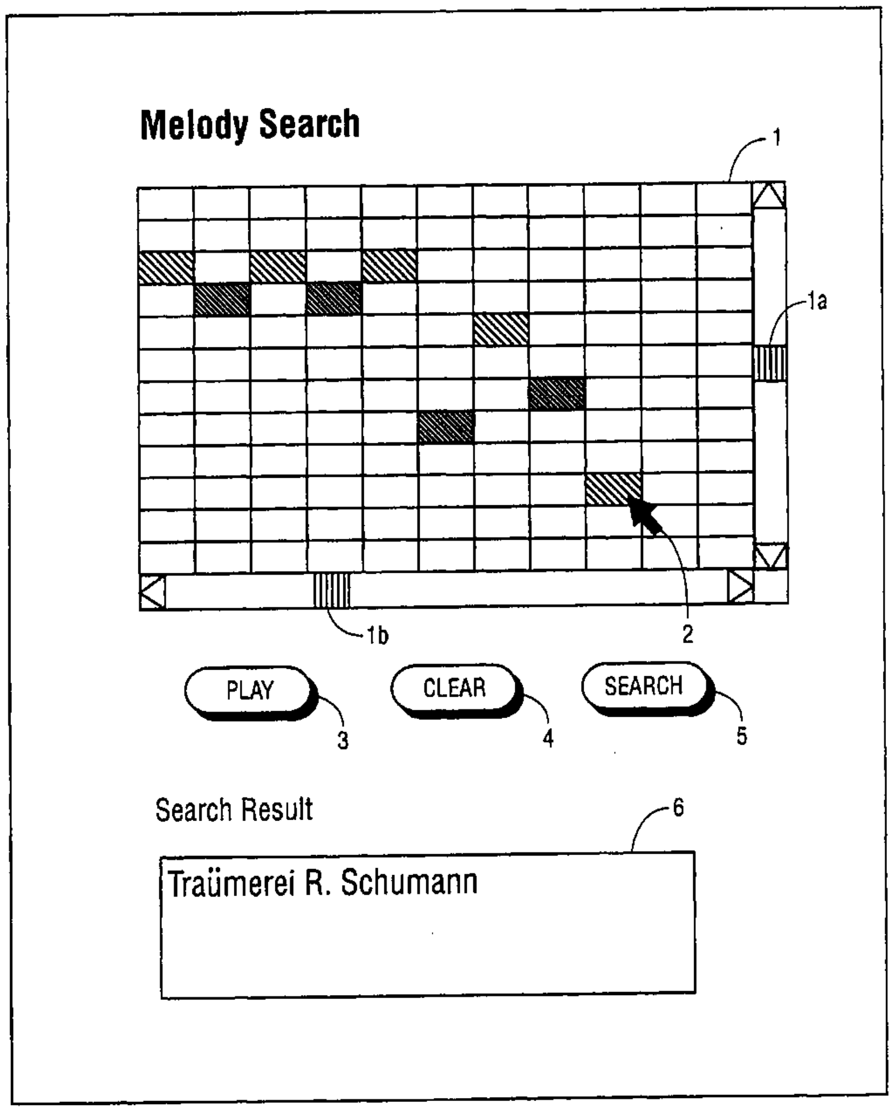


FIG. 2

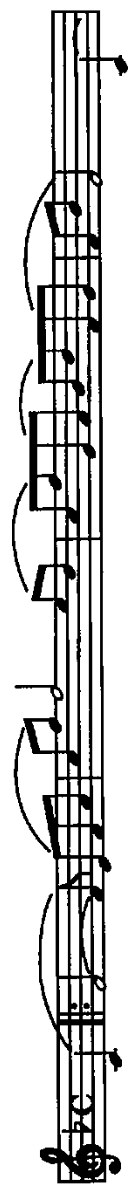
Für Elise

L.v. Beethoven

FIG. 3

Träumerei

R. Schumann, Op.15, No.7



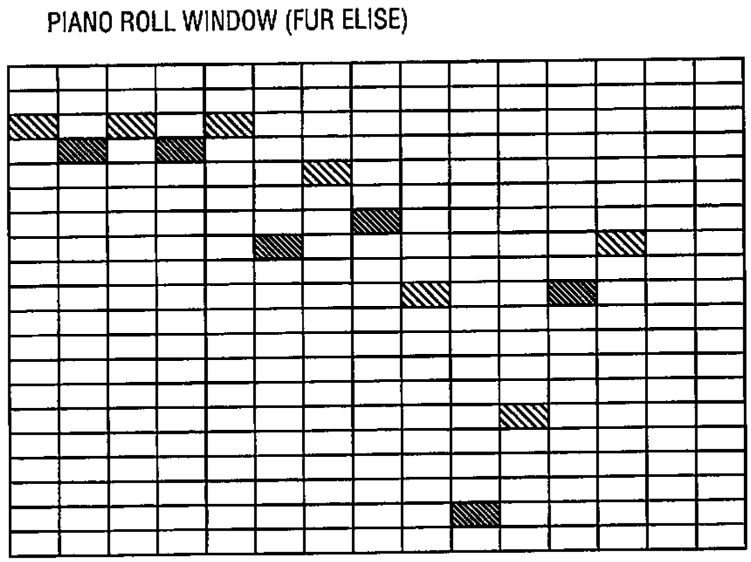


FIG. 4

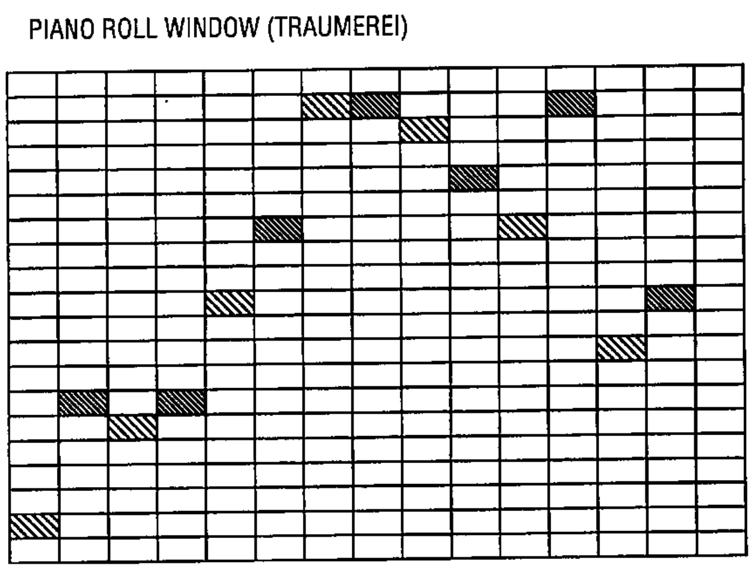


FIG. 5

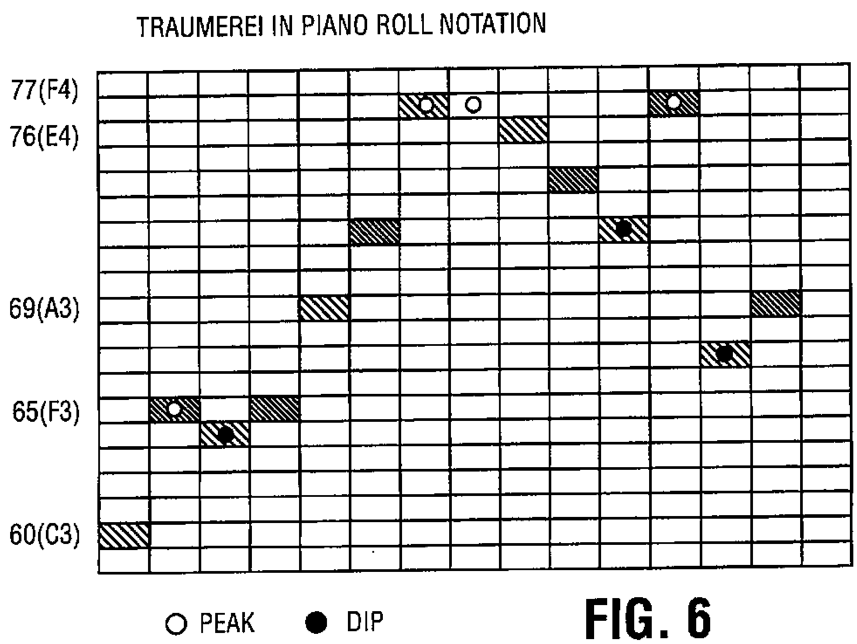

TRAUMEREI IN PIANO ROLL NOTATION
77(F4)
76(E4)
69(A3)
65(F3)
60(C3)
PEAK
DIP

FIG. 6

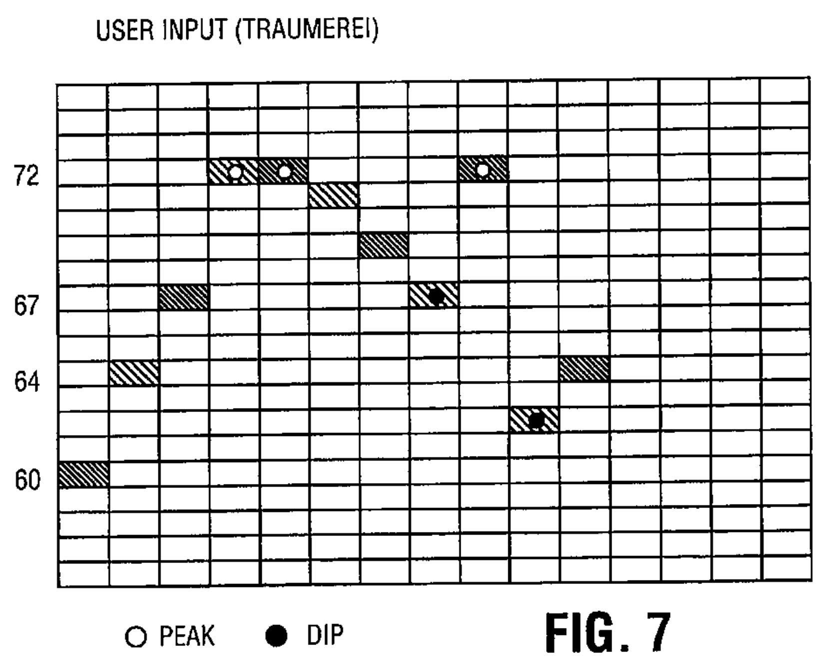

USER INPUT (TRAUMEREI)
72
67
64
60
PEAK
DIP

FIG. 7

FIG. 8

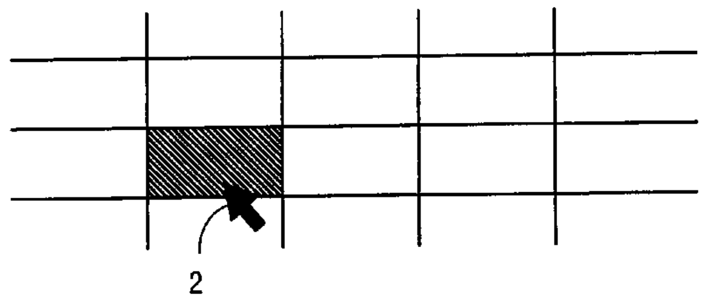


FIG. 9

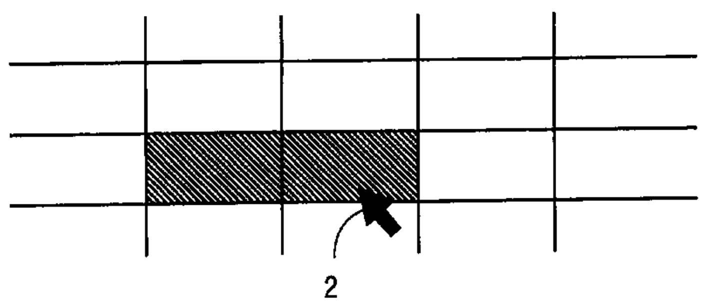
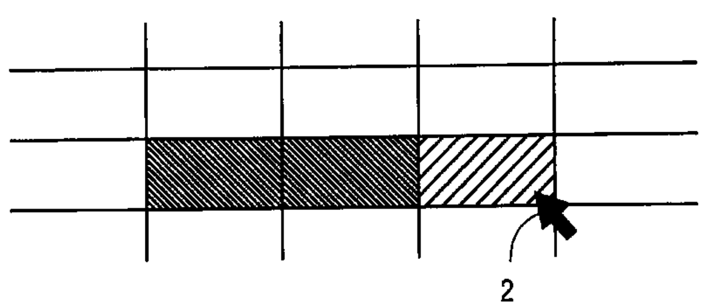


FIG. 10



TRAUMEREI IN PIANO ROLL NOTATION

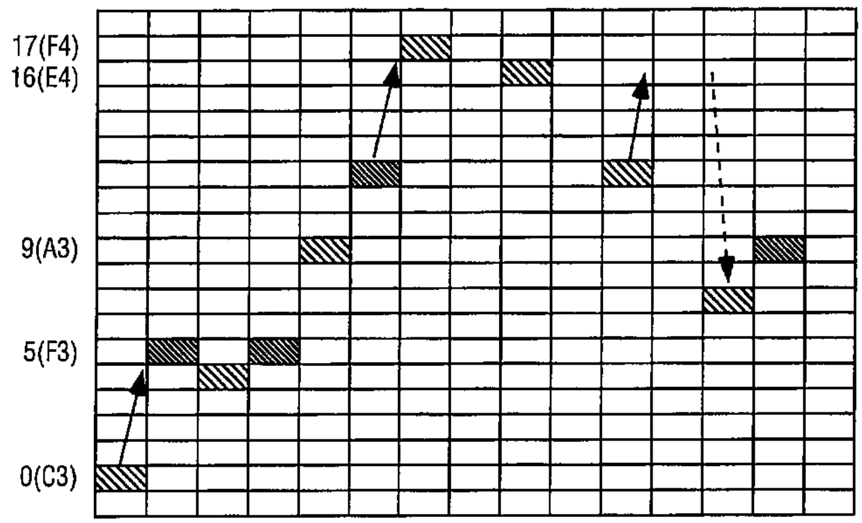


FIG. 11

USER INPUT (TRAUMEREI)

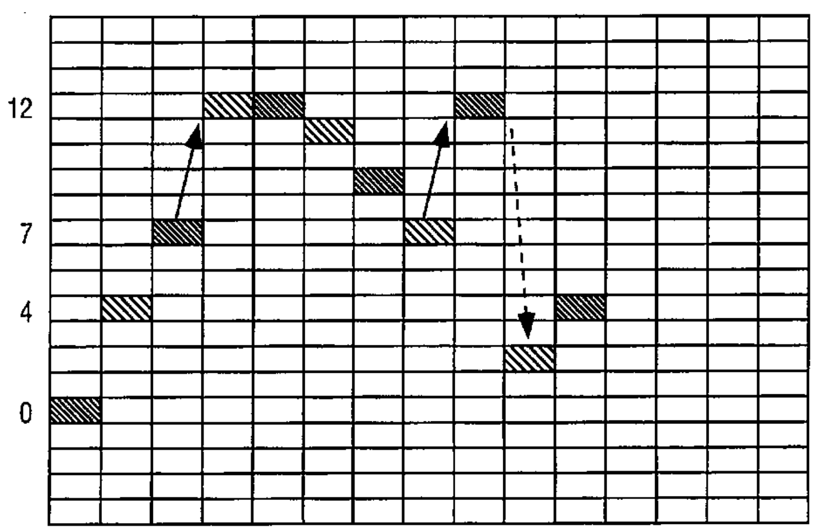
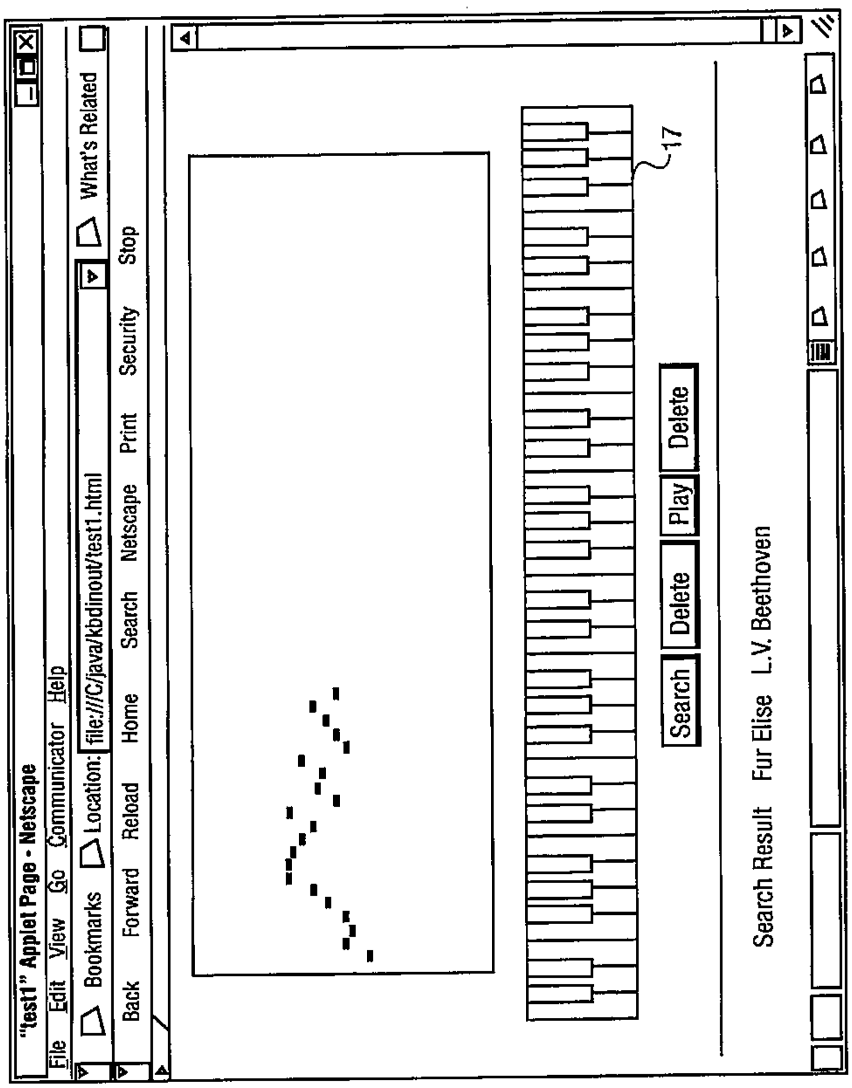


FIG. 12

Google Ex. 1012

FIG. 13


"test1" Applet Page - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: file://C:/java/kbdinout/test1.html
What's Related
Back Forward Reload Home Search Netscape Print Security Stop
17
Search
Delete
Play
Delete
Search Result Fur Elise L.V. Beethoven

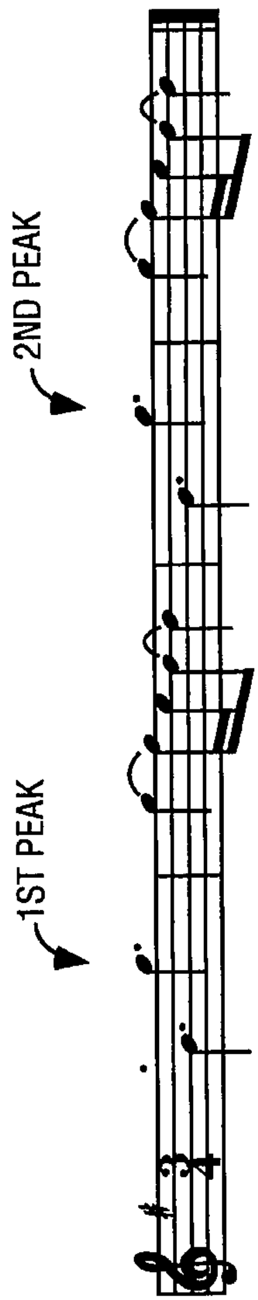


FIG. 14

MUSIC SEARCH BY MELODY INPUT

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention is directed to a mechanism for enabling the searching of songs by their melodies in sound files such as midi files.

2. Description of the Related Art

In the prior art, if one knows a song melody, but not the artist or title, about the only way to determine the title would be to ask a person having knowledge of the melody and title such as a person who works in a retail record/compact disc store. Existing music search capability available utilizing the Internet usually requires at least a part of its title or composer's name.

In U.S. Pat. No. 5.739.451. a hand held music reference device is described with song segments, typically stored as compressed MIDI files. as well as titles, lyrics, artists and the like. However, searching is generally performed by entering at least one of the title. lyrics, artist or the like which is then searched. Reference is made to note structure comparator 62 and performing a melody line search utilizing a note structure entered via directional keys. A note structure is defined as a series of relative note or pitch values. i.e., a melody line which is rising, falling or remaining the same in pitch value. Comparator 62 operates to locate songs which have the inputted note structure. Apparently, comparator 62 is a microprocessor which has been programmed to perform this function, the details of which are not set forth as searching by note structure does not appear to be the preferred mechanism for performing a search. Also, the entering of a note structure as described in this patent provides a limited mechanism for enabling the comparator to locate a desired song since basing a search on rising, falling or remaining the same pitch values provides only limited information.

In U.S. Pat. No. 5,402,339, a more sophisticated mechanism is described for performing a search based on a melody. However, to perform the search, a user must enter a string of note data items which is a period and scale level of a single sound identified by a musical note. To do this, the user must possess a high level of music knowledge and sophistication, making the described mechanism unsuitable for use by the average person having a limited knowledge of music. Further, period, i.e., note duration is very difficult to estimate, even for a user having a high level of music knowledge.

SUMMARY OF THE INVENTION

A method and apparatus is disclosed to enable one to search for a song title when only its melody is known. The invented music search allows a user to search a database and thereby obtain the title of the work only with its melody as input to a search engine and a minimal knowledge of music. The invention uses a piano roll music notation interface or a piano keyboard interface which allows the user to enter the melody in an easy to understand manner. This invention assumes a user has access to a remote music database through, for example, the Internet using a personal computer (PC) which functions as a client, with a PC keyboard, monitor, preferably a sound card, optionally a microphone and Internet or other network connection. Upon receiving a search request, the database server i.e., a remote computer, which can be accessed by the user via the client PC, sends a web pages for example, in HTML or Java containing the search results to the client PC. The client PC receives and displays the search results on the monitor and exchanges data with the server. Sound may also be played through the sound card to connected speakers.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a piano roll grid onto which a melody may be entered.

FIG. **2** is a portion of the work entitled Für Elise in standard music notation.

FIG. **3** is a portion of the work entitled Träumerei in standard music notation.

FIG. **4** illustrates a piano roll window for the melody (Für Elise) shown in FIG. **2**.

FIG. **5** illustrates a piano roll window for the melody (Träumerei) shown in FIG. **3**.

FIG. **6** illustrates Traumerei in piano roll notation.

FIG. **7** illustrates user input for Träumerei in piano roll notation.

FIG. **8** illustrates a portion of the piano roll window showing the selection of a particular block representing a note.

FIG. **9** illustrates a portion of the piano roll window showing the selection of a particular block representing a note with a duration twice as long as the note shown in FIG. **8**.

FIG. **10** illustrates a portion of the piano roll window showing the selection of a particular block representing a note twice as long as the note shown in FIG. **8** with a following note with the same pitch.

FIG. **11** illustrates user input for Träumerei in piano roll notation.

FIG. **12** illustrates user input for Träumerei in piano roll notation.

FIG. **13** illustrates an alternate interface for inputting a melody using a representation of a piano keyboard.

FIG. **14** illustrates a melody having a repeated pattern

DETAILED DESCRIPTION OF THE INVENTION

FIG. **1** shows an example of a search interface according to the present invention. The interface includes piano roll grid **1**, mouse or other pointing device cursor **2**, play button **3**, clear button **4**, search button **5**, and search result window **6**. Piano roll grid **1** has vertical scroll button **1***a* and horizontal scroll button **1***b*. Even if a user does not know traditional musical notation, with this piano roll input, the user can easily input a melody to search. This kind of music input interface is already used in existing music notation software. For example, a commercially available product known as Midisoft Studio 6.0 utilizes a piano roll input interface. See, http://www.midisoft.cornmupgrade.htm. In this case however, the interface is used to create and edit music rather than to perform searches.

The horizontal axis of the grid represents time and the vertical axis of the grid indicates pitch. Each block represents a note. One block above represents a half note. If the current block is C, its upper block is C#.

The user moves cursor **2** to a block to select it and then presses the mouse button. If the selected box pitch is C4, a tone of C4 will be generated by the soundcard of the PC and the user hears the tone. This tone feedback allows the user to correct the pitch easily. If the pitch is correct, the user releases the mouse button and that block is fixed. If the pitch is wrong, the user moves cursor 2 upward or downward to find the correct pitch. To correct a fixed block, the user moves the cursor onto the block and drags-and-drops it onto another block to correct the pitch. By dragging cursor 2 to the right, the user can prolong the duration of a note. For example, in FIG. 8, a block is chosen by moving cursor 2 onto it. In FIG. 9, by continuing to press the mouse button, the user drags cursor 2 to the next rightmost block and releases the button. These two blocks, which are solid, represent one note with duration 2. In FIG. 10, the third block is clicked. Because the mouse button has been released, this block, which is hatched, is detected as a following note with the same pitch. In this manner, the user inputs each note in the melody to search.

When the user finishes the melody input, play button 3 is clicked. The whole melody which has been entered is generated by the soundcard in the PC and is heard through connected speakers. If the melody is correct, search button 5 is selected to start the search. If the user wants to delete all entries made in the piano roll grid, clear button 4 is clicked.

In this search system, the duration of each note is ignored because it is difficult, if not impossible, for a user to input the correct duration. No rests (period of silence) are considered, either. For this reason, the user does not have to be concerned about the duration of each note. However, since it is easier for some users to enable each note to have at least an approximate duration to playback their input melody, if the user desires to input duration. such capability is provided. However, duration information is not used to perform a search.

FIG. 2 is the famous melody in "Für Elise" by L. V. Beethoven. FIG. 4 shows an example of piano roll notation for "Für Elise". FIG. 3 is another example melody. "Tr äumerei" by R. Schumann. FIG. 5 is the piano roll notation for this melody. The user may input the melody in any key. The invented search system automatically shifts its key to fit the data in the database as described in detail below. The user need only be concerned with pitch distance between each note.

For those who are familiar with music notation, there are alternative input methods. A piano keyboard and a music sheet are displayed as shown in FIG. 13. The user chooses a key on keyboard 17 by pointing to the desired key with a cursor and clicking a mouse button. A corresponding note appears on the sheet 19. This kind of traditional input method may alternatively be employed. In this case, if a MIDI instrument is available. for example, a MIDI keyboard, it may be connected to the PC, and the user may enter a melody from the keyboard.

The invented keyboard (or piano arid) interface is designed for on-line access. In order to generate a tone of each piano key, the network server sends the sound files to the client computer over the network. The client computer does not have to store the sound files. They will be sent on demand in the preferred embodiment. There are several choices for sound files. One is a MIDI data file. The MIDI file has only a pitch and duration values. So its size is very compact. It can be accessed over a network relatively quickly. A disadvantage is that the client computer must have a MIDI sound generator. Most UNIX computers have no MIDI generator. Thus, although a MIDI file is small, using MIDI limits the number of clients that can be used. Another solution is a waveform file like a .au or .wav file which are well known sound file formats. Each waveform file is several Kbytes. 60-key data is about 250 kbytes. It is larger than a MIDI FIle and thus takes more time to send. However, no MIDI sound generator is required. Compared with a MIDI file, more clients can handle the waveform files.

Also, instead of a keyboard or piano grid input interface, a microphone can be used. The microphone receives the user's voice (e.g., by humming) and converts it to an electronic signal. The signal is amplified and analog-to-digital converted. The digitized signal is analyzed in a CPU by a FFT (Fast Fourier Transform) algorithm according to well known techniques in which FFT is used to analyze sound by obtaining frequency spectrum information from waveform data. The frequency of the voice is obtained and finally a musical note that has the nearest pitch is selected.

This kind of interface is already in practical use. For example, the following web pages describe software products with such an interface.

http://www.medianavi.co.jp/product/hana/tk9907.html
http://www.medianavi.co.jp/product/hana/step.html
http://www.medianavi.co.jp/product/hana/step2.html
http://www.medianavi.co.jp/product hana/step3.html

Regardless of the input method employed, in the preferred embodiment. a user uses a browser, such as Netscape Navigator available from America on Line installed on the client computer and performs the following steps to do a search.

(1) The client requests the search engine web page (e.g,. Java applet) over a network (e.g., the Internet).
(2) The web server receives the request and sends back the web page data which includes the user interface described above, which is for example a Java applei and sound files.
(3) T he client computer displays the web page. The user enters the melody to search using any of the above-described techniques or anv other suitable mechanism to enter the melody. The client computer sends the melody data to the web server.
(4) The web server passes the melody data to a CGI (Common Gateway Interface) script. The CGI script is, for example, a Perl (Practical Extraction Report Language) script, an example of which follows:

```
# Melody Search Program
if($ENV{QUERY_STRING}) {
    $entered = SENV{QUERY_STRING};
}
@search_result = 'melody_search.exe @melody';
print "Content-type: text/html\n\n";
print "<html>\n";
print "<head>\n";
print "<title>Melody Search Result</title>\n";
print "</head>\n";
print "<body>";
print "<H2>Melody Search Result </H2>\n";
print "<hr>";
if($ENV{QUERY_STRING}) {
        print "@search_result \n";
}
print "<hr>";
print "</body>\";
print "</html>\n";
```

(5) The CGI script calls the search program (the invented peak search program described below) and gives the melody data to it.
(6) The search program searches the database and obtains the information about the entered melody.
(7) The search program returns the search result to the CGI script.
(8) The CGI script passes the result to the web server.

(9) The web server returns the search result to the client. The client browser displays it.

Most of the existing online search engines have a similar search process. The search engine as described is novel at least as to the following points.

(a) The web server provides a Java applet and sound files. So, the user can enter a melody while listening to play-back. The user can use any Java-ready computer.

(b) Fast Peak search. The peaks in all the melodies stored in the databases are marked in advance. For melody matching, the entered melody is time-shifted, as explained below, so that its peak matches each peak in the reference melody.

The Music Melody Database

A composer's name and a music title are stored with its melody data in the database. If necessary, related information, for example, performers, history, background, can be linked to the database.

The melody data may be stored as a MIDI file. When searched, its melody is extracted from the MIDI file and compared with the input melody. If search speed does not matter, this is a good solution because the user listens to the searched music by playing the MIDI file. The melody stored in the database could be not only the beginning of the melody, but also one or more famous phrases in the middle of the music.

The following is an example of a database structure for storing the foregoing information in C language notation. Each melody data is grouped with its composer ID and its title ID as follows. Here, up to 64 notes are stored for a melody. Of course, the size of the array used to store the melody may be adjusted up or down as desired, depending on the amount of storage available.

```
struct melody_database {
        int composer_ID;
        int title_ID;
        int m[64];
}
```

Thus, a music database for that has 5000 classical melodies would be defined as follows:

struct melody_database DB[5000];

Key Shift and Pattern Matching

Composer-ID list and Title-ID list are shown below. DB[0] is assigned for "Träumerei".

3650 is stored in DB[0].composer_ID. DB[0].title_ID is 1720. The search engine finds that the entered melody matches DB[0] and obtains these two Ids. Then they are referred to the lists and the composer's name "Schumann" and the title "Träumerei" are obtained.

Composer-ID list

ID	Composer
3600	SCHEIN, Johann Hermann (1586–1630)
3610	SCHLICK, Arnolt (c. 1460–c. 1517)
3620	SCHOENBERG, Arnold (1874–1951)
3630	SCHUBERT, Franz Peter (1797–1828)
3640	SCHUBERT, Franz (1808–78)
3650	SCHUMANN, Robert Alexander (1810–56)
3660	SCHUTZ, Heinrich (1585–1672)
3670	SCRIABIN, Alexander (1872–1915)
3680	SERMISY, Claudin de (c.1490–1562)
3690	SGAMBATI, Giovanni (1841–1914)

-continued

Title-ID list

ID	Title
1710	Toccata, Op. 7
1720	Träumerei (Kinderszenen Op. 15)
1730	Fantasia in C, Op. 17
1740	Arabeske, Op. 18
1750	Novelettes No. 6, Op. 21
1760	Romance No. 2, Op. 28

A pitch number is assigned to each key as shown in Table 1.

TABLE 1

Key	Value	Key	Value	Key	Value	Key	Value
C1	36	C2	48	C3	60	C4	72
C #1	37	C #2	49	C #3	61	C #4	73
D1	38	D2	50	D3	62	D4	74
D #1	39	D #2	51	D #3	63	D #4	75
E1	40	E2	52	E3	64	E4	76
F1	41	F2	53	F3	65	F4	77
F #1	42	F #2	54	F #3	66	F #4	78
G1	43	G2	55	G3	67	G4	79
G #1	44	G #2	56	G #3	68	G #4	80
A1	45	A2	57	A3	69	A4	81
A #1	46	A #2	58	A #3	70	A #4	82
B1	47	B2	59	B3	71	B4	83

Peak Search

Träumerei starts with C3. which has value 60. The first 21 notes are represented as A[0][0 . . . 21]={60, 65, 64, 65, 69, 72, 77, 77, 76, 74, 72, 77, 67, 69, 70, 74, 65, 67, 69, 72, 67}.

Note that these absolute pitch values need not be stored in the database. A difference between two adjacent notes is stored to DB[ ].m[ ]. (When no index is given to an array m[ ], it indicates the entire data of the array.) If the database has absolute pitch data, an entered melody must be key-shifted to each melody in the database. This is very time-consuming. To avoid key-shift, absolute pitch data is converted to relative pitch data and used for matching. Relative pitch data is obtained by the next formula.

Absolute pitch data: $a[i]$

where $0 \le i < m$ and m is the length of the melody.

Relative pitch data: $r[i] = a[i+1] - a[i]$ where $0 \le i < m-1$.

Träumerei's relative pitch data is stored from DB[0].m[0] to DB[0].m[20].

DB[0].m[0 . . . 20]={5, −1, 1, 4, 3, 5, 0, −1, −2, −2, 5, −10, 2, 1, 4, −9, 2, 2, 3, −5}.

In DB[0].m[21] and later, fill data such as 9999, is stored. The length of melody in the database is flexible, which in the example, is up to 64 bytes. It should be long enough to uniquely identify the melody.

Peak notes are also detected and marked when the database is built. A peak note is defined as the note that is higher than or equal to each of the adjacent notes. When both of left and right note are equal to the note, i.e. when three consecutive notes have the same pitch, the center (second) note is not regarded as a peak. In FIGS. 6 and 7, a white dot indicates a peak note. In relative pitch notation, if the next value is positive or zero and the next value is negative or zero, the current value is marked as a peak. In case of a series of zeroes, only the first zero is marked. In the next representation, a peak is marked with an asterisk.

DB[0].m[0 . . . 20]={*5, -1, 1, 4, 3, *5, *0, -1, -2, -2, *5, -10, 2, 1, 4, -9, 2, 2, 3, -5}.

Similarly, the absolute pitch data of Für Elise is represented as

A[1][0 . . . 16]={76, 75, 76, 75, 76, 71, 74, 72, 69, 60, 64, 69, 71, 64, 68, 71, 72}. DB[1].m[0 . . . 15] will be DB[1].m bx;1[0 . . . 15]={-1, *1, -1, *1, -5, *3, -2, -2, -9, 4, 5, *2, -7, 4, 3, 1}.

Assume that the user inputs a part of Träumerei into the piano roll grid as shown in FIG. 7. The user may enter a melody in any key. The first note in FIG. 7 is C3, which is 60. (The actual pitch is F3.) Note that the user does not always enter a melody from the beginning correctly. In this case, the user has dropped the first three notes. The pitch data of the entered melody is as follows.

Ain[0 . . . 10]={60, 64, 67, 72, 72, 71, 69, 67, 72, 62, 64}.

The relative pitch data of Ain[ ] will be

Rin[0 . . . 11]={4, 3, *5, *0, -1, -2, -2, *5, -10, 2}.

Next, Rin[ ] is time-shifted for matching. Rin[ ] will be compared to DB[0].m[ ]. Since the first peak in Rin[ ] is Rin[2].,Rin[ ] is time-shifted (i.e, moving the array to the left or right, as appropriate) so that Rin[2] locates in the position of DB[0].m[0].

{*5, -1, 1, 4, 3, *5, *0, -1, -2, -2, *5, -10, 2, 1, 4, -9, 2, 2, 3, -5}{4, 3, *5, 0, -1, -2, -2, *5, -10, 2}.

0 1 2 6 5 0 10 3

Each value below the line is an absolute difference of the two associated values. The values in un-overlapped area are neglected. The total absolute difference is

0+1+2+6+5+0+10+3=27.

When the two melodies completely match, the total absolute difference will be zero. The goal is to find a reference melody that gives the least total absolute difference. Again, Rin[ ] is time-shifted (in this case, moved to the right) so that Rin[2] and DB[0].m[5] match.

{*5, -1, 1, 4, 3, *5, *0, -1, -2, -2, *5, -10, 2, 1, 4, -9, 2, 2, 3, -5}{4, 3, *5, 0, -1, -2, -2, *5, -10, 2}.

0 0 0 0 0 0 0 0 0 0

This time, the total absolute difference is zero. The two melodies completely match. DB[0] is the melody the user is looking for. DB[0].composer_ID and DB[0].title_ID are returned as a search result. The result is indicated in search result window 6 in FIG. 1. In this manner, the entered melody is shifted to each peak in each reference melody and compared. The reference melody that gives the least difference is returned as a search result.

To accelerate the search, computation of the total absolute difference can be stopped when it exceeds a certain limit.

Linked Features

If this database is linked to a music database that has complete music score data, searched music may be played automatically. For example, DB[0].title_ID may be linked to the MIDI file of Träumerei . The server sends the MIDI file to the client PC and the client PC play it. Moreover, the result can be linked to the list Compact Disc on on-line CD shop. For example, all the CDs that includes Träumerei are shown on the display. By clicking one of them, it is sent to the shopping cart of on-line purchaser.

Variations

Instead of a peak, a dip can be used. A dip note is indicated with a black dot in FIGS. 6 and 7.

Also both peaks and dips may be used together. The number of peaks and dips in each reference melody are detected beforehand and a flag indicates which are fewer in number. Before matching, this flag is detected and it is decided which is to be used, peaks or dips. If dips are fewer than peaks, the entered melody is shifted so that the dip in the entered melody locates in the position of the dip in the reference melody. This method will save computation time because search time greatly depends on the number of matching peaks or dips required to be made.

Moreover, in order to further accelerate the search, the order of comparison may be considered for search speed improvement. It is most probable that the highest peak in the entered melody matches the highest one in the reference melody. Therefore, the highest peak is first compared with the highest one in the reference melody and next compared with the second highest peak. For example, in Träumerei, there are four peaks, DB[0].m[0], DB[0].m[5], DB[0].m[6], and DB[0].m[10]. Their absolute pitch value are A[0][1], A[0][6], A[0][7], A[0][11]. They are respectively, 65, 77, 77, and 77. The highest peak in Träumerei is DB[0].m[5], DB[0].m[6], and DB[0].m[10]. DB[0].m[0] is the second highest. The height order is stored in the database beforehand. The highest peak in the entered melody Rin[ ] is Rin[2], Rin[3], and Rin[7] are the highest and they are 72. For matching, the entered melody is shifted so that the Rin[2] locates in the same position of the highest peak DB[0].m[5], DB[0].m[6] and DB[0].m[10] respectively. After this match, the entered melody is shifted to the second highest peak DB[0].m[0].

Instead of using a peak , a differential between notes may also be used. The largest step-up in the input melody is detected. In relative pitch data DB[ ].m[ ], the largest value indicates the largest step-up. In case of Träumerei, DB[0].m[15] is the largest step-up, which is marked with a pound. The second largest step-up is marked with a double-pound. In this way, each step-up in the database is numbered in large step-up order.

DB[0].m[0 . . . 20]={##5, -1, 1, 4, 3, ##5, 0, -1, -2, -2, ##5, -10, 2, 1, 4, #9, 2, 2, 3, 5}

Each step-up in the entered melody is also marked in the same way when it is entered.

Rin[0 . . . 11]={4, 3, #5, 0, -1, -2, -2, #5, -10, 2}.

The entered melody is shifted so that the largest entered step-up locates at the largest step-up in the reference melody. The absolute difference between the two melodies are computed as described above. If the largest step-up is done, the entered melody is shifted to the second largest step-up (marked with a double-pound) in the reference melody. In this way, the entered melody is shifted to each step-up and compared. FIGS. 11 and 12 illustrate this step-up search. These figures have the same pattern as FIGS. 6 and 7 respectively. Solid arrows indicate the largest step-up in FIG. 12. It is five steps. Instead of step-up, step-down data can be used. In FIGS. 11 and 12, a dotted arrow indicates the largest step-down.

Instead of relative pitch data, the original absolute pitch data can be stored in the database and used for matching. The input melody is key-shifted so that the peaks in the input melody have the same pitch as each peak in the reference melody. This is done by comparing peaks in the input melody and reference melody, and then subtracting the difference from each subsequent note until the next peak, and then repeating for the second and each subsequent peak. A disadvantage of absolute pitch data is that a key-shift is required for every matching. An advantage is original absolute pitch data can be stored without modification. Also a MIDI file, which includes pitch, duration, and other data, can be used although it takes time to analyze MIDI format.

Advantages of Peak Search

(1) Search Speed

Peak notes are approximately 20% of the total number of notes in a typical melody. That means search speed using peak notes is 20% of a brute force search which shifts the entered melody, note by note. Also because relative pitch data is used, no key-shift is required to compare with each reference melody.

(2) No Restriction on Incomplete Input

The entered melody is shifted based on a peak note in the melody. Therefore, the user can start a melody with any note. In the above example, an exact result was obtained even though the first three notes of the melody were dropped . The only restriction is that an entered melody must include at least one peak.

(3) Input Fault Tolerance

The user does not always enter the melody without mistakes. Some notes could be dropped or given a wrong pitch. So a search engine should have some input fault tolerance. If a peak note is correctly entered, this search engine will find the closest melody from the database. Even if a peak has a wrong pitch, another peak can be used for the search. In the above example, assume Rin[2] has the wrong pitch. In this case, no exact result is obtained. So, another search will be done with the second peak Rin[7]. When Rin[7] is shifted to DB[0].m[10], the total absolute difference will be a minimum and the correct result will be obtained. (As described above, a dip can be used instead of a wrong peak.) For these reasons, a correct search can be obtained notwithstanding inaccurate input from the user.

(4) Flexibility on Search Area

Some melodies have a repeated pattern as shown in FIG. 14. In this melody, the second measure is identical to the first one. Two peaks are in the melody, but the second peak is not necessary to test because the user also enters the same repeated pattern. As described above, each peak in the database is marked when the database is built. In this peak-marking process, each repeated peak can be omitted. This omission avoids unnecessary searching and accelerates search speed. Also, a peak that is in an unimportant portion can be skipped. In a long music selection, there are some important portions that are indispensable to identify the melody. These portions are well recognized and remembered by the user. The user identifies such important portions as a keyword (key-melody). The other unimportant portions can often be ignored. Therefore, peaks in an unimportant portion can be omitted in the same way as a repeated peak. This omission also contributes to performing a fast search. Boyer-Moore (discussed below) or other string-matching algorithms do not have this kind of flexibility. They only search word by word from the beginning of the database to the end.

Alternative Search Method

A musical note can be represent as an integer like an ASCII character. Therefore, a melody data can be handled in the same manner as a character string. Fundamentally, a melody search is equivalent to string search. Of course, however, a key-shift is required for a melody search. There are many studies for fast and efficient string search techniques. For example, the Boyer-Moore algorithm is well-known as one of the best solutions. See

http://orca.st.usm.edu/~suzi/

http://www-igm.univ-mlv.fr/~mac/DOC/B5-survey.html#pm.

Instead of the using these techniques to perform a string search, these search algorithms may be applied to perform melody searches.

Further Search With A Wildcard

When a regular search does not result in a good match, further searches using wildcards can be performed. In a string search, a wildcard like "?" is used instead of an uncertain character. The character "?" matches any character. For example, if it desired to find a three-letter word which starts with "a" and ends with "d", the search keyword entered would be "a?d". As a search result, "add", "aid", "and", etc. would be returned. Such a wildcard can be introduced to music search according to the present invention. A problem is that the user may not know what note of the entered melody is wrong. So, it is desirable that a search engine automatically searches with the modified input melody. The invented search engine also has such input fault tolerance capability.

First, in the case when string-search algorithm is used, assume that the user entered n notes as follows.

(a[1], a[2], a[3] . . . a[n])

The index of the array starts with 1 for simplicity. (Note that actual search uses not a[ ], but a relative difference between two adjacent notes.)

(1) Correction of a Wrong Pitch

The search engine assumes one of the n notes has a wrong pitch. The search engine replaces one of them with a wildcard and tries a search. There are n variations. The database is searched n times. The modified input melody is represented as follows.

(?, a[2], a[3], . . . a[n])

(a[1], ?, a[3], . . . a[n])

. . .

(a[1], a[2] . . . a[n−1], ?)

(2) Compensation for a Dropped Note

The search engine assumes one note has been dropped. The search engine adds a wildcard as a new note for further search. There are n+1 variations. The database is searched n+1 times. The modified input melody is represented as follows.

(?, a[1], a[2], a[3], . . . a[n])

(a[1], ?, a[2], a[3], . . . a[n])

. . .

(a[1], a[2], a[3], . . . a[n], ?)

(3) Removal of a Redundant Note

The search engine assumes one of the n notes is redundant. The search engine drops a note and tries further search. This has n variations. The search is repeated n times. The modified input melody is represented as follows.

(a[2], a[3], . . . a[n])

(a[1], a[3], . . . a[n])

. . .

(a[1], a[2], . . . a[n−1])

Totally, the further search takes 3n+1 times longer than the regular search. The invented method can be applied for two or more erroneous notes in the same way. If computational speed is fast enough, further search for two or more errors can be done.

For the peak search, more consideration is required. Depending on what note is modified, a decision must be made as to which peak is to be used for the search.

(4) Correction of a Wrong Pitch

Assume that a[2] is the original peak note. An asterisk indicates a peak.

(a[1], *a[2], a[3], . . . a[n])

The search engine assumes one of the n notes has a wrong pitch. It replaces one of them with a wildcard. When a note to be replaced is neither the peak nor the two adjacent notes, the peak will be still a peak after modification. Since a peak is defined only by two adjacent notes, modification to the other notes does not affect the original peak. So, the search will be done using the original peak. For example, even if a[n−1] is replaced with a wildcard, a[2] is still a peak. So, the modified melody can be shift so that a[2] locates at each peak of the reference melodies.

(a[1], *a[2], a[3], . . . a[n−2], ?, a[n])

In case a note to be replaced is either the original peak or the two adjacent notes. When another peak exists and the modification to these three notes does not affect that peak, it is possible to use it as a substitute peak. In the next example, let a[n−1] be another peak.

(a[1], *a[2], a[3], . . . a[n−2], *a[n−1], a[n])

Even after one of a[1] to a[3] is modified, a[n−1] is still a peak. So, a[n−1] will be used. What if such another peak does not exist? The original peak is used for the search although it might not be a peak after modification. (This is the worst case scenario.)

(5) Compensation for a Dropped Note

The search engine assumes one note has been dropped and adds a new note as a wildcard. This is similar to (4). If the newly added note is placed outside of the original peak and the two adjacent notes., the original peak is used. In the next example, the new notes are placed between a[n−1] and a[n]. That never affects a[1] to a[3]. So, search will be done using a[2].

(a[1], *a[2], a[3], . . . a[n−1], ?, a[n])

In case that a wildcard is placed next to the peak, if available, another peak is used. In the next examples, another peak a[n−2] is used for search.

(a[1], ?, *a[2], a[3], . . . a[n−1], *a[n−2], a[n])

(a[1], *a[2], ?, a[3], . . . a[n−1], *a[n−2], a[n]])

If no other peak is available, a[2] is used.

(6) Removal of a Redundant Note

The solution for this case is simple. After removing a redundant note, a new peak is found in the modified input melody. The new peak is used for the search.

The invented input fault tolerance function allows the user to obtain an exact result even when an entered melody has some errors. The user does not have to use a wildcard. When a regular search does not obtain a good match, the input melody is automatically modified with a wildcard and further search starts with the modified melody.

Conclusions

The invented melody search as described above has the following advantages.

A user can search a music title from its melody. No other information is required to be input.

The melody input is easy. No traditional music notation is required.

The user does not have to pay attention to the duration of each note. Only the pitch is used for the search.

A user can playback the input melody to verify, its correctness before performing the search.

This search engine is designed not only for stand-alone PC but also for on-line access over the network. The interface software is written in multi-platform language like Java. So a user can access the remote server with any computer. (Windows PC, Macintosh, UNIX, etc.)

Interface software including sound files are sent to the client from the server on demand. The client do not have to install any files in advance.

A fast search is performed by using a peak or differential matching algorithm.

The flexible nature of the search algorithm enables the searching of an incomplete melody. It has input fault tolerance.

When you build the database, by un-marking peaks you can select the portions that should not be searched. This avoids searching unnecessary portions and accelerates search speed.

The search result can be linked to its sound (MIDI) file. The user can immediately listen to its music.

The search result can be linked to on-line music shop. By clicking a CD list, the user can easily purchase the CD that includes the music.

The invented melody search provides a very useful, user-friendly and fast mechanism to search music. A user can obtain an exact music title from a melody and use the information to, for example, purchase its CD or tape.

I claim:

1. A method for searching for a musical piece which includes at least one melody comprising the steps of:

a) receiving note information representing the melody provided by a user;

b) converting the received information to a series of values corresponding to each received note information;

c) calculating relative pitch values from said series of values

d) comparing said relative pitch values to values stored in said database representing stored melodies;

e) selecting as the musical piece being searched one of said stored melodies which produces a closest match based on said comparing.

2. The method defined by claim 1 further comprising the step of marking pitch peaks in said stored values and determining peak pitch values from said relative pitch values for said comparison step.

3. The method defined by claim 2 further comprising the step of time shifting said relative pitch values to match a first peak in said relative pitch values with a first peak in said stored values.

4. The method defined by claim 3 further comprising the step of time shifting said relative peak values to match subsequent peaks in said relative pitch values with corresponding subsequent peaks in said stored values.

5. The method defined by claim 4 wherein the closest match is determined by calculating absolute differences between said time shifted relative peak values and said stored values, said closest match being one have a smallest calculated absolute difference.

6. The method defined by claim 1 further comprising the step of marking pitch dips in said stored values and determining dip pitch values from said relative pitch values for said comparison step.

7. The method defined by claim 5 further comprising the step of time shifting said relative pitch values to match a first dip in said relative pitch values swith a first dip in said stored values.

8. The method defined by claim 7 further comprising the step of time shifting said relative dip values to match subsequent dips in said relative pitch values with corresponding subsequent dips in said stored values.

9. The method defined by claim 8 wherein the closest match is determined by calculating absolute differences between said time shifted relative dip values and said stored values, said closest match being one have a smallest calculated absolute difference.

10. The method defined by claim 1 further comprising the step of marking largest pitch differentials in said stored values and determining largest differential pitch values from said relative pitch values for said comparison step.

11. The method defined by claim 10 further comprising the step of time shifting said relative pitch values to match a first largest difference in said relative pitch values with a first largest difference in said stored values.

12. The method defined by claim 11 further comprising the step of time shifting said relative dip values to match subsequent largest differences in said relative pitch values with corresponding subsequent largest differenes in said stored values.

13. The method defined by claim 12 wherein the closest match is determined by calculating absolute differences between said time shifted relative largest difference values and said stored values, said closest match being one have a smallest calculated absolute difference.

14. The method defined by claim 1 wherein said note information is received from a user based on a piano roll grid interface in which cells within an array represent pitch values and are selected by a user using a pointing device.

15. The method defined by claim 1 wherein said note information is received from a user based on a piano keyboard grid interface in which keys on a piano keyboard represent pitch values and are selected by a user using, a pointing device.

16. The method defined by claim 1 wherein said note information is received from a user based upon sounds input to a microphone representing the melody.

17. The method defined by claim 1 farther comprising the step of playing said note information for verification of its correctness.

18. The method defined by claim 1 wherein note information includes at least one wildcard character.

19. The method defined by claim 1 further comprising the step of playing the selected musical piece.

20. The method defined by claim 1 further comprising the step of enabling the user to make an on-line purchase of the selected musical piece.

21. A method for searching for a musical piece which includes at least one melody comprising the steps of:

a) receiving note information representing the melody provided by a user;

b) converting the received information to a series of values corresponding to each received note information;

c) adjusting said series of values by key shifting said values;

d) comparing said key shifted pitch values to absolute pitch values stored in said database representing stored melodies;

e) selecting as the musical piece being searched one of said stored melodies which produces a closest match based on said comparing.

22. The method defined by claim 21 further comprising the step of marking pitch peaks in said stored values and determining peak pitch values from said key shifted pitch values for said comparison step.

23. The method defined by claim 22 further comprising the step of time shifting said key shifted pitch values to match a first peak in said relative pitch values with a first peak in said stored values.

24. The method defined by claim 23 further comprising the step of time shifting said key shifted peak values to match subsequent peaks in said relative pitch values with corresponding subsequent peaks in said stored values.

25. The method defined by claim 24 wherein the closest match is determined by calculating absolute differences between said time shifted and key shifted peak values and said stored values, said closest match being one have a smallest calculated absolute difference.

26. The method defined by claim 21 further comprising the step of marking pitch dips in said stored values and determining dip pitch values from said key shifted pitch values for said comparison step.

27. The method defined by claim 21 further comprising the step of time shifting said key shifted pitch values to match a first dip in said key shifted pitch values with a first dip in said stored values.

28. The method defined by claim 27 further comprising the step of time shifting said key shifted dip values to match subsequent dips in said key shifted pitch values with corresponding subsequent dips in said stored values.

29. The method defined by claim 28 wherein the closest match is determined by calculating absolute differences between said time shifted and key shifted dip values and said stored values, said closest match being one have a smallest calculated absolute difference.

30. The method defined by claim 21 further comprising the step of marking largest pitch differentials in said stored values and determining largest differential pitch values from said key shifted pitch values for said comparison step.

31. The method defined by claim 30 further comprising the step of time shifting said key shifted pitch values to match a first largest difference in said key shifted pitch values with a first largest difference in said stored values.

32. The method defined by claim 31 further comprising the step of time shifting said key shifted dip values to match subsequent largest differences in said key shifted pitch values with corresponding subsequent largest differenes in said stored values.

33. The method defined by claim 32 wherein the closest match is determined by calculating absolute differences between said time shifted and key shifted largest difference values and said stored values, said closest match being one have a smallest calculated absolute difference.

34. The method defined by claim 21 wherein said note information is received from a user based on a piano roll grid interface in which cells within an array represent pitch values and are selected by a user using a pointing device.

35. The method defined by claim 21 wherein said note information is received from a user based on a piano keyboard grid interface in which keys on a piano keyboard represent pitch values and are selected by a user using a pointing device.

36. The method defined by claim 21 wherein said note information is received from a user based upon sounds input to a microphone representing the melody.

37. The method defined by claim 21 further comprising the step of playing said note information for verification of its correctness.

38. The method defined by claim 21 wherein note information includes at least one wildcard character.

39. The method defined by claim 21 further comprising the step of playing the selected musical piece.

40. The method defined by claim 21 further comprising the step of enabling the user to make an on-line purchase of the selected musical piece.

* * * * *

Query: *5 -1 1 4 3 *5 0 -1 -1 -2 -2

Reference: 4 3 *5 0 -1 -2 -2 *5 -10 2

Comparisons:

	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8	Position 9	Position 10
Comparison No. 1	-2	-	-	-	-	-	-	-	-	-
Comparison No. 2	-2	-2	-	-	-	-	-	-	-	-
Comparison No. 3	-1	-2	-2	-	-	-	-	-	-	-
Comparison No. 4	-1	-1	-2	-2	-	-	-	-	-	-
Comparison No. 5	0	-1	-1	-2	-2	-	-	-	-	-
Comparison No. 6	*5	0	-1	-1	-2	-2	-	-	-	-
Comparison No. 7	3	*5	0	-1	-1	-2	-2	-	-	-
Comparison No. 8	4	3	*5	0	-1	-1	-2	-2	-	-
Comparison No. 9	1	4	3	*5	0	-1	-1	-2	-2	-
Comparison No. 10	-1	1	4	3	*5	0	-1	-1	-2	-2
Comparison No. 11	*5	-1	1	4	3	*5	0	-1	-1	-2
Comparison No. 12	-	*5	-1	1	4	3	*5	0	-1	-1
Comparison No. 13	-	-	*5	-1	1	4	3	*5	0	-1
Comparison No. 14	-	-	-	*5	-1	1	4	3	*5	0
Comparison No. 15	-	-	-	-	*5	-1	1	4	3	*5
Comparison No. 16	-	-	-	-	-	*5	-1	1	4	3
Comparison No. 17	-	-	-	-	-	-	*5	-1	1	4
Comparison No. 18	-	-	-	-	-	-	-	*5	-1	1
Comparison No. 19	-	-	-	-	-	-	-	-	*5	-1
Comparison No. 20	-	-	-	-	-	-	-	-	-	*5
	4	3	*5	0	-1	-2	-2	*5	-10	2

Query: *5 4 3 2 1

Reference: *6 2 2 2 2 1 *6 5 4 3 2 1

Comparisons:

	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8	Position 9	Position 10	Position 11	Position 12	Total Absolute Difference
Comparison No. 1	1	-	-	-	-	-	-	-	-	-	-	-	5
Comparison No. 2	2	1	-	-	-	-	-	-	-	-	-	-	5
Comparison No. 3	3	2	1	-	-	-	-	-	-	-	-	-	4
Comparison No. 4	4	3	2	1	-	-	-	-	-	-	-	-	4
Comparison No. 5	*5	4	3	2	1	-	-	-	-	-	-	-	5
Comparison No. 6	-	*5	4	3	2	1	-	-	-	-	-	-	6
Comparison No. 7	-	-	*5	4	3	2	1	-	-	-	-	-	12
Comparison No. 8	-	-	-	*5	4	3	2	1	-	-	-	-	15
Comparison No. 9	-	-	-	-	*5	4	3	2	1	-	-	-	15
Comparison No. 10	-	-	-	-	-	*5	4	3	2	1	-	-	12
Comparison No. 11	-	-	-	-	-	-	*5	4	3	2	1	-	5
Comparison No. 12	-	-	-	-	-	-	-	*5	4	3	2	1	0
Comparison No. 13	-	-	-	-	-	-	-	-	*5	4	3	2	4
Comparison No. 14	-	-	-	-	-	-	-	-	-	*5	4	3	6
Comparison No. 15	-	-	-	-	-	-	-	-	-	-	*5	4	6
Comparison No. 16	-	-	-	-	-	-	-	-	-	-	-	*5	4
	*6	2	2	2	2	1	*6	5	4	3	2	1	

Query: 5 4 3 2 1

Reference: 0 0 5 4 3 2 1 0 0 5 4 3 2 1 0 0 5 4 3 2 1 0 0 [illegible] 0 0

US005874686A

United States Patent [19]

Ghias et al.

[11] Patent Number: 5,874,686

[45] Date of Patent: Feb. 23, 1999

[54] **APPARATUS AND METHOD FOR SEARCHING A MELODY**

[76] Inventors: **Asif U. Ghias**, 2306 Sultana Dr., Yorktown Heights, N.Y. 10598; **Jonathan A. Logan**, 1186 Palomino Rd., Santa Barbara, Calif. 93105; **David W. Chamberlin**, 206 Flynn Ave., Mountain View, Calif. 94043

[21] Appl. No.: **742,260**

[22] Filed: **Oct. 31, 1996**

Related U.S. Application Data

[60] Provisional application No. 60/008,177, Oct. 31, 1995.

[51] **Int. Cl.**[6] **A63H 5/00**; G04B 13/00; G10H 7/00

[52] **U.S. Cl.** **84/609**; 84/616

[58] **Field of Search** 84/600, 609, 634, 84/616, 654

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,688,464	8/1987	Gibson et al. .	
4,771,671	9/1988	Hoff, Jr.	84/645
5,038,658	8/1991	Tsuruta et al. .	
5,088,380	2/1992	Minamitaka	84/637
5,402,339	3/1995	Nakashima et al. .	
5,510,572	4/1996	Hayashi et al.	84/609
5,619,004	4/1997	Dame .	

FOREIGN PATENT DOCUMENTS

405061917A	3/1993	Japan .

OTHER PUBLICATIONS

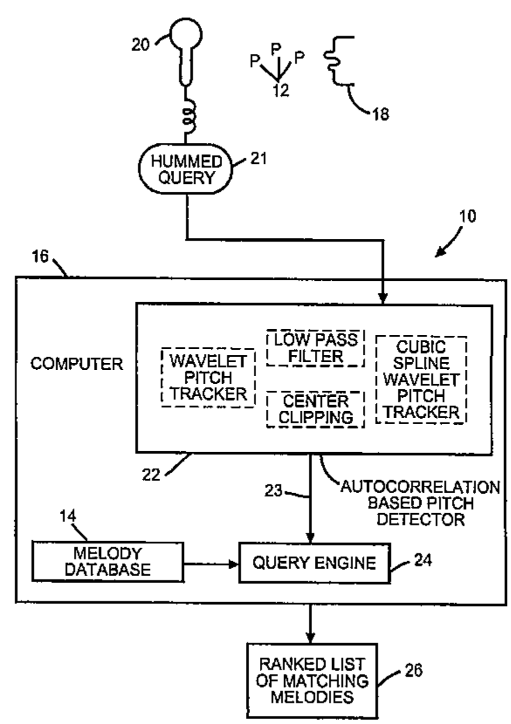
Hawley, Michael J., Structure out of Sound, 1993, MIT Doctoral Thesis, Abstract.

Primary Examiner—William M. Shoop, Jr.
Assistant Examiner—Jeffrey W. Donels
Attorney, Agent, or Firm—Hodgson, Russ, Andrews, Woods & Goodyear LLP

[57] **ABSTRACT**

Apparatus and method for easily, efficiently, and accurately searching melodies. A melody is inputted to a computer and converted into a form of a sequence of digitized representations of relative pitch differences between successive notes thereof. A melody database is searched for at least one sequence of digitized representations of relative pitch differences between successive notes which at least approximately matches the sequence of digitized representations of relative pitch differences between successive notes of the melody.

16 Claims, 3 Drawing Sheets



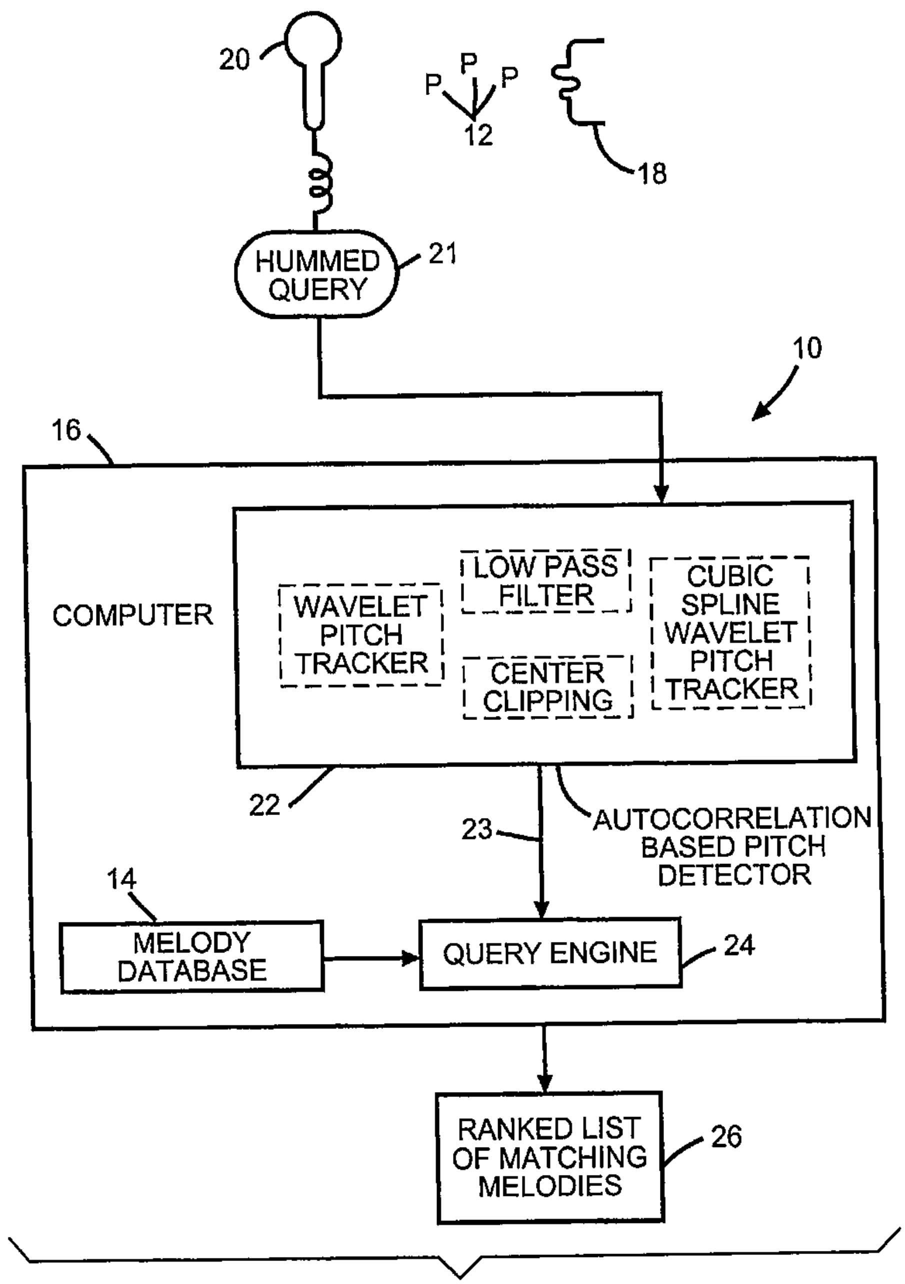

20
P
P
P
12
18
HUMMED QUERY
21
10
16
COMPUTER
WAVELET PITCH TRACKER
LOW PASS FILTER
CENTER CLIPPING
CUBIC SPLINE WAVELET PITCH TRACKER
22
23
AUTOCORRELATION BASED PITCH DETECTOR
14
MELODY DATABASE
QUERY ENGINE
24
RANKED LIST OF MATCHING MELODIES
26

FIG.1

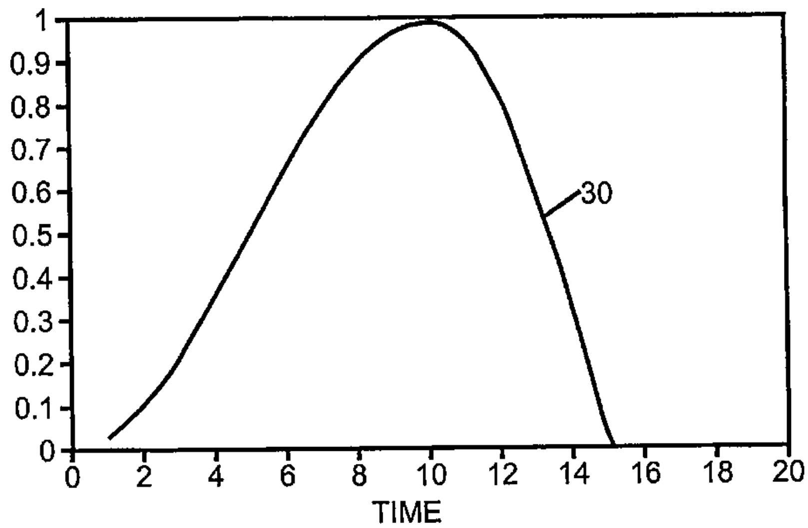

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
30
0
2
4
6
8
10
12
14
16
18
20
TIME

FIG. 2

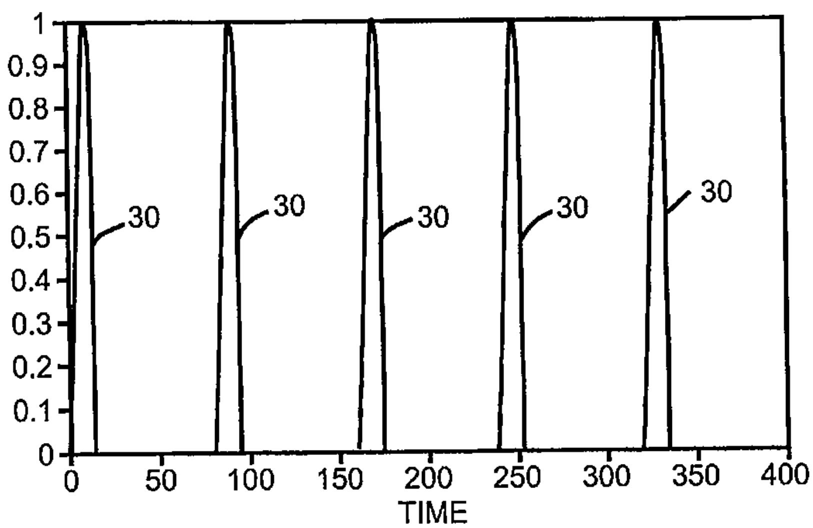

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
30
30
30
30
30
0
50
100
150
200
250
300
350
400
TIME

FIG. 3

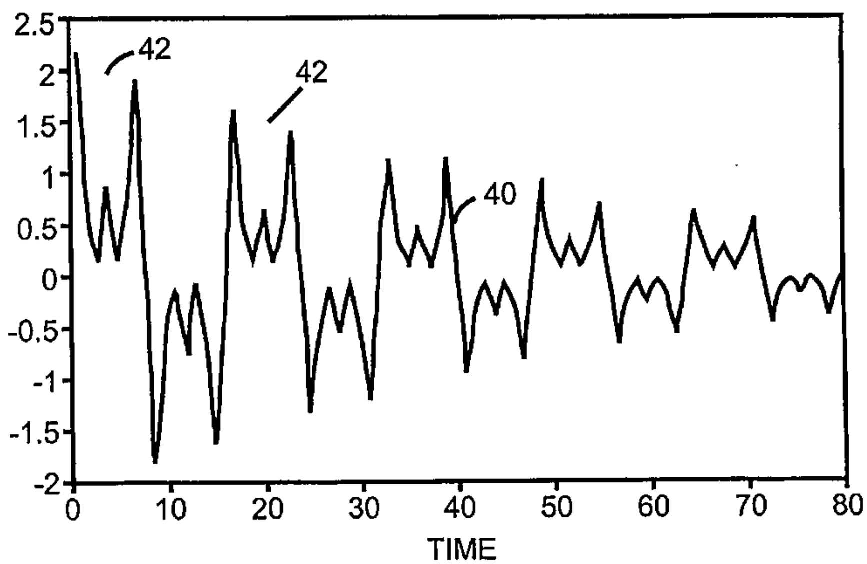

2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
42
42
40
0
10
20
30
40
50
60
70
80
TIME

FIG. 4

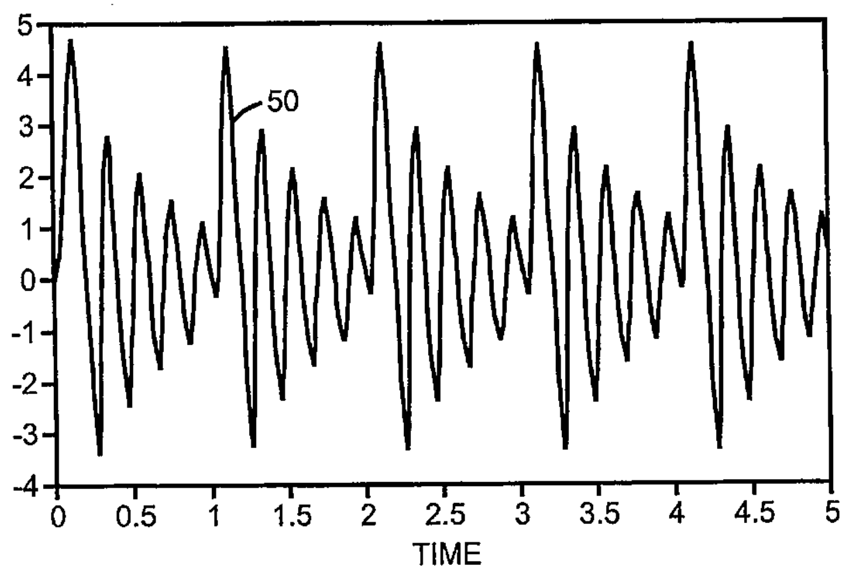

5
4
3
2
1
0
-1
-2
-3
-4
50
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
TIME

FIG. 5

APPARATUS AND METHOD FOR SEARCHING A MELODY

CROSS REFERENCE TO A RELATED APPLICATION

This application claims priority of provisional application Ser. No. 60/008,177, filed Oct. 31, 1995, which is hereby incorporated herein by reference.

BACKGROUND OF THE INVENTION

The present invention relates generally to database searching. More particularly, the present invention relates to melody searching.

Next generation databases should include image, audio, and video data in addition to traditional text and numerical data. These data types will require query methods that are more appropriate and natural to the type of respective data. For instance, a natural way to query an image database is to retrieve images based on operations on images or sketches supplied as input. Similarly, a natural way of querying an audio database (of songs) is to hum the tune of a song, as apparently addressed in T. Kageyama and Y. Takashima, "A Melody Retrieval Method With Hummed Melody" (language: Japanese), *Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, J77D-II(8): 1543–1551, August 1994. Such a system would be useful in any multimedia database containing musical data by providing an alternative and natural way of querying. One can also imagine a widespread use of such a system in commercial music industry, music radio and TV stations, music stores, and even for one's personal use.

It has been observed that melodic contour, defined as the sequence of relative differences in pitch between successive notes, can be used to discriminate between melodies. See Stephen Handel, *Listening: An Introduction to the Perception of Auditory Events*, The MIT Press, 1989, which indicates that melodic contour is one of the most important methods that listeners use to determine similarities between melodies. In Michael Jerome Hawley, *Structure out of Sound*, PhD thesis, MIT, September 1993, a method of querying a collection of melodic themes by searching for exact matches of sequences of relative pitches input by a MIDI keyboard is briefly discussed.

U.S. Pat. No. 5,510,572 discloses utilizing pitch differences between successive notes in classifying motion for a melody analyzer and harmonizer, wherein a search may be incidentally used to find an appropriate chord progression to, for example, harmonize music so as to accompany a singer. Other art which may be of interest includes U.S. Pat. Nos. 5,040,081; 5,146,833; 5,140,886; 4,688,464, and 5,418,322.

SUMMARY OF THE INVENTION

It is an object of the present invention to easily, efficiently, and accurately search melodies.

In order to easily, efficiently, and accurately search melodies, in accordance with the present invention, a computer means is provided which has a database of melodies each including a plurality of notes in a form of a sequence of digitized representations of relative pitch differences between successive notes, a melody is inputted to the computer means and converted into a form of a sequence of digitized representations of relative pitch differences between successive notes thereof, and the melody database is searched for at least one sequence of digitized representations of relative pitch differences between successive notes which at least approximately matches the sequence of digitized representations of relative pitch differences between successive notes of the melody.

The above and other objects, features, and advantages of the present invention will be apparent in the following detailed description of a preferred embodiment thereof when read in conjunction with the accompanying drawings wherein the same reference numerals denote the same or similar parts throughout the several views.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating an apparatus and method which embodies the present invention.

FIGS. **2** and **3** are graphs illustrating an excitation signal and train of such excitation signals respectively used to create a synthesized pitch for the method and apparatus.

FIG. **4** is a graph of a formant structure which is formed from certain formant frequencies.

FIG. **5** is a graph of a synthesized pitch created by convolving the train of excitation pulses of FIG. **3** and the formant structure of FIG. **4**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. **1**, there is illustrated generally at **10** apparatus for searching a melody, which is represented by a series of successive notes, illustrated at **12**, which of course have different pitches. In other words, it may be desirable to know the identity of the particular melody or tune **12**, and a database, illustrated at **14**, of melodies or tunes is searched, as described hereinafter, to locate at least one melody or tune which at least approximately matches the tune **12**. The database **14** is shown to be contained within a general purpose computer **16**, but, alternatively, the database **14** may be located apart from the computer **16** and suitably connected thereto for communicating between the computer and database. Both of these alternatives are meant to come within the scope of the present invention.

In accordance with the present invention, the tune **12** is hummed by a person **18** into a microphone **20**, and the hummed query, illustrated at **21**, is suitably digitized, in accordance with principles commonly known to those of ordinary skill in the art to which this invention pertains, and the digitized signals of the hummed query **21** are fed to a pitch tracking module **22** in computer **16**. The pitch tracker assembles a contour representation of the hummed melody **12**, as hereinafter discussed in greater detail, which is fed to a query engine, illustrated at **24**. The query engine **24** searches the melody database **14** and outputs a ranked list of approximately matching melodies, as illustrated at **26**. A preselected error tolerance may be applied to the search. The query engine **24** may of course alternatively be programmed to output the single most approximate matching melody or, if desired, to output an exact matching melody. However, by searching for an approximate matching melody, as hereinafter discussed, various forms of anticipated errors may be taken into account.

The database **14** of melodies may be acquired, for example, by processing public domain MIDI songs, or may otherwise be suitably acquired, and may be stored as a flat-file database. Pitch tracking may be performed in, for example, Matlab software, a product of the Matworks, Inc. of Natick, Mass., chosen for its built-in audio processing capabilities and the east of testing a number of algorithms within it. Hummed queries may be recorded in a variety of formats, depending upon the platform-specific audio input capabilities of the Matlab software. For example, the format may be a 16-bit, 44 Khz WAV format on a Pentium system, or a 8-bit 8 Khz AU format on a Sun Sparcstation. The query engine **24** may use an approximate pattern matching algorithm, described hereinafter, in order to tolerate humming errors.

User input **12** (humming) to the system **10** is converted into a sequence of relative pitch transitions, as follows. A note **12** in the input may be classified in one of three ways: a note is either the same as the previous note (S), higher than the previous note (U), or lower than the previous note (D). Thus, the input is converted into a string with a three letter alphabet (U,D,S). For example, the introductory theme of Beethoven's 5th Symphony would be converted into the sequence: — S S D U S S D (the first note is ignored as it has no previous pitch). For another example, the melody portion **12**, as illustrated in FIG. **1** with one high note between two lower notes, would be converted into the sequence: U D.

To accomplish this conversion, a sequence of pitches in the melody must be isolated and tracked. This is not as straight-forward as it sounds, however, as there is still considerable controversy over exactly what pitch is. The general concept of pitch is clear: given a note, the pitch is the frequency that most closely matches what one hears. Performing this conversion in a computer can become troublesome because some intricacies of human hearing are still not clearly understood. For instance, if one plays the 4th, 5th, and 6th harmonics of some fundamental frequency, one actually hears the fundamental frequency, not the harmonics, even though the fundamental frequency is not present. This phenomenon was first discovered by Schouten in some pioneer investigations carried out from 1938 to 1940. Schouten studied the pitch of periodic sound waves produced by an optical siren in which the fundamental of 200 Hz was cancelled completely. The pitch of the complex tone, however, was the same as that prior to the elimination of the fundamental. See R. Plomp, *Aspects of Tone Sensation,* Academic Press, London, 1976.

Because of the interest in tracking pitch in humming, methods were examined for automatically tracking pitch in a human voice. Before one can estimate the pitch of an acoustic signal, we must first understand how this signal is created, which requires forming a model of sound production at the source. The vibrations of the vocal cords in voiced sounds are caused as a consequence of forces that are exerted on the laryngeal walls when air flows through the glottis (the gap between the vocal cords). The terms "vocal folds" and "vocal cords" are more or less used as synonyms in the literature. Wolfgang Hess, *Pitch Determination of Speech Signals,* Springer-Verlag, Berlin Heidelberg, 1983, describes a model of the vocal cords as proposed by M. Hirano, *"Structure and Vibratory Behavior of the Vocal Folds,"* in M. Sawashima and F. S. Cooper, editors, *Dynamic aspects of speech production,* pages 13–27, University of Tokyo Press, 1976. For the purposes of the present application though, it is sufficient to know that the glottis repeatedly opens and closes thus providing bursts of air through the vocal tract.

The vocal tract can be modeled as a linear passive transmission system with a transfer function H(z). If one adds an additional transfer function R(z) which takes into account the radiation, the output impedance of the vocal tract can approximately be set to zero. In the neutral position where the vocal tract can be regarded as a uniform tube, the resonances of the vocal tract occur at sound wavelengths of

$$\lambda = \frac{4L}{(2k-1)c}\ ; \quad k = 1, 2, 3, \ldots$$

With L–17 cm (average value of vocal-tract length) and a sound propagation speed of c=340 meters/sec., the frequencies of these resonances will be:

$$F_k = (2k-1)\cdot 500 \text{ Hz};\ k=1,2,3, \ldots$$

The frequencies F_k are called formant frequencies.

The resulting sound that is heard is considered to be the convolution of the excitation pulse, illustrated at **30** in FIG. **2**, created by the glottis and the formant frequencies. Therefore, if one wants to model a speech signal, one starts with a train of excitation pulses **30**, as shown in FIG. **3**. The period T_o in the train of excitations **30** is 0.01 sec. making the pitch 100 Hz. For the formant frequencies, the above equation for F_k is used with $k \in \{1,2,3\}$. This gives formant frequencies: F_1=500 Hz, F_2=1500 Hz, and F_3=2500 Hz. Combining these frequencies and adding an exponential envelope produces the formant structure shown at **40** in FIG. **4**. By convolving the train of excitation pulses **30** with the formant structure **40** a synthesized pitch, as shown at **50** in FIG. **5**, is obtained.

Since what one hears as pitch is considered to actually be the frequency at which the bursts of air occur, one may be able to find the pitch of the segment by tracking the bursts of air.

There are at least three approaches to tracking pitch.

The Maximum Likelihood approach, which is described by James D. Wise, James R. Caprio, and Thomas W. Parks in "Maximum Likelihood Pitch Estimation," *IEEE Trans. Acoust., Speech, Signal Processing,* 24(5): 418–423, October 1976, is a modification of autocorrelation (hereinafter discussed) that increases the accuracy of the pitch and decreases the chances of aliasing.

However, this approach is very slow due to its computational complexity. An implementation in Matlab software may typically take approximately one hour or longer to evaluate 10 seconds of audio on a 90 MHz Pentium workstation. With some optimizations, the performance may be improved to approximately 15 minutes per 10 seconds of audio, but this is still far too slow for the purposes of this invention.

The Cepstrum Analysis approach, which is discussed in A. V. Oppenheim, "A Speech Analysis-Synthesis System Based on Homomorphic Filtering," *J. Acoustical Society of America,* 45: 458–465, February 1969, and in Alan V. Oppenheim and Ronald W. Schafer, *Discretetime-Signal Processing,* Prentice Hall, Englewood Cliffs, N.J., 1989, does not give very accurate results for-humming.

In order to provide both rapid and accurate pitch tracking, a preferred pitch tracking approach is autocorrelation, which is described in L. R. Rabiner, J. J. Dubnowski, and R. W. Schafer, "Realtime Digital Hardware Pitch Detector," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-24(1): 2–8, February 1976. This approach isolates and tracks the peak energy levels of the signal which is a measure of the pitch. Referring back to FIG. **4**, the signal s(n), illustrated at **40**, peaks where the impulses occur. Therefore, tracking the frequency of these peaks **42** should provide the pitch of the signal. In order to get the frequency of these peaks **42**, autocorrelation is employed, which is defined by:

$$R(l)=\sum_{k=-\infty}^{\infty} h(k)h(l+k)$$

where l is the lag and h is the input signal.

While autocorrelation is subject to aliasing (picking an integer multiple of the actual pitch) and is computationally complex, it still provides greater speed and accuracy as compared to the previously discussed approaches. The present implementation of autocorrelation was found to require approximately 45 seconds for 10 seconds of 44 KHz, 16-bit audio on a 90 MHz pentium workstation.

In order to improve the performance and speed and robustness of the pitch-tracking algorithm, a cubic-spline wavelet transform or other suitable wavelet transform may be used. The cubic spline wavelet peaks at discontinuities in the signal (i.e., the air impulses). One of the most significant features of the wavelet analysis is that it can be implemented in a filter bank structure and therefore computed in 0(n) time.

It is preferred that the cubic spline wavelet implementation be an event-based implementation of the tracer.

The cubic spline wavelet transform is linear and shift-invariant which are useful properties for speech signals since they are often modelled as a linear combination of shifted and damped sinusoids.

If a signal x(t) or its derivatives have discontinuities, then the modulus of the cubic spline transform of x(t) exhibits local maxima around the points of discontinuity. This is considered to be a desirable property for a cubic spline wavelet pitch-tracker since glottal closure causes sharp changes in the derivative of the air flow in the glottis and transients in the speech signal.

It has been shown in S. Mallat and W. L. Hwang, "Singularity Detection and Processing with Wavelets", Technical Report, Courant Institute of Mathematical Sciences, New York, March, 1991, that if one chooses a wavelet function g(t) that is the first derivative of a smoothing function (a smoothing function is a function whose Fourier transform has energy concentrated in the low-frequency region) P(t), then the local maxima of the transform indicate the sharp variations in the signal whereas the local minima indicate the slow variations. Hence, the local maxima of the transform using a wavelet, which is the first derivative of a smoothing function, is considered to be desirable for detecting the abrupt changes or transients in the speech signal caused by glottal closure.

One of the most appealing features of the cubic spline wavelet transform is that is can be implemented in a filter bank structure. It therefore is 0(n) fast.

Preferably, the autocorrelation pitch tracking approach includes low-pass filtering (for removing non-vocal frequencies) and center-clipping, as described in M. M. Sondhi, "New Methods of Pitch Extraction," *IEEE Trans. Audio Electroacoust. (Special Issue on Speech Communication and Processing—Part II)*, AU-16: 262–266, June 1968. It is considered that these will help eliminate the formant structure that generally causes difficulty for autocorrelation based pitch detectors.

This preferred form of autocorrelation takes between 20 and 45 seconds on a Sparc10 workstation to process typical sequences of hummed notes. A brute-force search of the database unsurprisingly shows linear growth with the size of the database, but remains below 4 seconds for 100 songs on a Sparc2. Therefore, the search time is currently effectively limited by the efficiency of the pitch-tracker.

In order to search the database, songs in the database **14** are preprocessed to convert the melody into a stream of the previously discussed U,D,S characters, and the converted user input (the key **23**) is compared with all the songs.

The pattern-matching desirably uses a "fuzzy" search to allow for errors within the matches. These errors reflect the inaccuracies in the way people hum as well as errors in the representation of the songs themselves.

For performing the key-search within the database **14**, it is considered desirable to use an efficient approximate pattern matching algorithm. By "approximate" is meant that the algorithm should be able to take into account various forms of errors.

Using the word CASABLANCA as an example, the following are examples of the types of errors which may occur in a typical pattern matching scheme.

	In CASABLANCA	In a melody
Transposition error	CASABALNCA	two notes reversed
Dropout error	CAS BLANCA	a note dropped
Duplication error	CASAABLANCA	a note duplicated
Insertion error	CASABTLANCA	a note inserted

Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from 0(mn) for the brute force algorithm to 0(kn) or 0(nlog(m), where "0" means "on the order of," m is the number of pitch differences in the query, and n is the size of the string (song). See Ricardo Baeza-Yates and G. H. Gonnet, "Fast String Matching with Mismatches," *Information and Computation*, 1992. A preferred algorithm which is considered to offer better performance in general for this purpose is that described in Ricardo A. Baesa-Yates and Chris H. Perieberg, "Fast and Practical Approximate String Matching, "*Combinatorial Pattern Matching, Third Annual Symposium*," pages 185–192, 1992.

This algorithm addresses the problem of string matching with k mismatches. The problem consists of finding all instances of a pattern string P=p1,p2,p3 . . . pm in a text string T=t1,t2,t3 . . . tn such that there are at most k mismatches (characters that are not the same) for each instance of P in T. When k=0 (no mismatches) one has the simple string matching problem, solvable in 0(n) time. When k=m, every substring of T of length m qualifies as a match, since every character of P can be mismatched. Each of the errors in the above CASABLANCA example corresponds to k=1.

The worst case for this algorithm occurs when P (the key **23**) consists of m occurrences of a single distinct character, and T (contour representation of song) consists of n instances of that character. In this case the running time is 0(mn). However, this is neither a common nor useful situation for the present invention. In the average case of an alphabet in which each character is equally likely to occur, the running time is

$$0\left(n\left(1+\frac{m}{|r|}\right)\right)$$

where |r| is the size of the alphabet.

The computer **16** may desirably be programmed so that, for a given query, the database **14** returns a list of songs ranked by how well they matched the query, not just one best match. The number of matches that the database **14** should retrieve depends upon the error-tolerance used during the key-search. This error-tolerance could be set in one of two possible ways: either it can be a user-definable parameter or the database can itself determine this parameter based on, for example, heuristics that depends on the length of the key. This would give the user an opportunity to perform queries even if the user is not sure of some notes within the tune.

From the results of the query the user can identify the song of interest. If the list is too large, the user can perform a new query on a restricted search list consisting of songs just retrieved. This allows the user to identify sets of songs that contain similar melodies.

An experimental evaluation of the system tested the tolerance of the system with respect to input errors, whether from mistakes in the user's humming or from problems with the pitch-tracking.

Effectiveness of the system **10** is directly related to the accuracy with which pitches that are hummed can be tracked and the accuracy of the melodic information within the database **14**. Under ideal circumstances, one can achieve close to 100% accuracy tracking humming, where "ideal circumstances" means that the user places a small amount of space between each note and hits each note strongly. For this purpose, humming short notes is encouraged. Even more ideal is for the user to aspirate the notes as much as possible, perhaps going so far as to voice a vowel, as in "haaa haaa haaa." Only male voices have been experimented with.

The evaluation database contained a total of 183 songs. Each song was converted from public domain General MIDI sources. Melodies from different musical genres were included, including both classical and popular music. A few simple heuristics were used to cut down on the amount of irrelevant information from the data, e.g., MIDI channel **10** was ignored as this is reserved for percussion in the General MIDI standard. However, the database still contained a great deal of information unrelated to the main theme of the melody. Even with this limitation, it is discovered that sequences of 10 to 12 pitch transitions were sufficient to discriminate 90% of the songs.

As a consequence of using a fast approximate string matching algorithm, search keys can be matched with any portion of the melody, rather than just the beginning. As the size of the database grows larger, however, this may not prove to be an advantage.

Contour representations for each song are stored in separate files. Opening and closing files may become a significant overhead. Performance may therefore be improved by packing all the songs into one file, or by using a database manager. The programming code may be modernized to make it independent of any particular database schema.

The pattern matching algorithm in the form previously described does not discriminate between the various forms of pattern matching errors discussed earlier, but only accounts for them collectively. Some forms of errors may be more common than others depending upon the way people casually hum different tunes. For example, drop-out errors reflected as dropped notes in tunes are more common than transposition or duplication errors. Tuning the key-search so that it is more tolerant to drop-out errors, for example, may yield better results.

The generation of the melodic contours of the source songs automatically from MIDI data is convenient but not optimal. More accuracy and less redundant information could be obtained by entering the melodic themes for particular songs by keyboard.

The resolution of the relative pitch differences may be desirably increased by using query alphabets of three, five, and more possible relationships between adjacent pitches. Early experiments using an alphabet of five relative pitch differences (same, higher, much higher, lower, much lower) indicated changes of this sort are promising. One drawback of introducing more resolution is that the users must be somewhat more accurate in the intervals they actually hum.

It should be understood that, while the present invention has been described in detail herein, the invention can be embodied otherwise without departing from the principles thereof, and such other embodiments are meant to come within the scope of the present invention as defined by the appended claims.

What is claimed is:

1. Apparatus for searching a melody comprising a computer means having a database of melodies each including a plurality of notes in a form of a sequence of digitized representations of relative pitch differences between successive notes, means for inputting a melody to said computer means, means for converting the melody into a form of a sequence of digitized representations of relative pitch differences between successive notes thereof, and means for searching said melody database for at least one sequence of digitized representations of relative pitch differences between successive notes which at least approximately matches said sequence of digitized representations of relative pitch differences between successive notes of the melody.

2. Apparatus according to claim 1 wherein said input means comprises a microphone.

3. Apparatus according to claim 1 further comprising means for outputting from said computer a ranked list of approximately matching melodies.

4. Apparatus according to claim 1 wherein said converting means comprises an autocorrelation based pitch detector means.

5. Apparatus according to claim 1 wherein said converting means comprises an autocorrelation based pitch detector means having low-pass filter means for removing non-vocal frequencies and center-clipping means for eliminating unwanted formant structure.

6. Apparatus according to claim 1 wherein said converting means comprises a wavelet pitch tracker.

7. Apparatus according to claim 1 wherein said converting means comprises a cubic spline wavelet pitch tracker.

8. Apparatus according to claim 1 further comprising means for approximately matching sequences of digitized representations of relative pitch differences between successive notes.

9. A method for searching a melody comprising the steps of: (a) selecting a computer means having a database of melodies each including a plurality of notes in a form of a sequence of digitized representations of relative pitch differences between successive notes, (b) inputting a melody to the computer means, (c) converting the melody into a form of a sequence of digitized representations of relative pitch differences between successive notes thereof, and (d) searching the melody database for at least one sequence of digitized representations of relative pitch differences between successive notes which at least approximately matches the sequence of digitized representations of relative pitch differences between successive notes of the melody.

10. A method according to claim 9 wherein the step of inputting the melody comprises humming into a microphone.

11. A method according to claim 9 further comprising operating the computer to output a ranked list of approximately matching melodies.

12. A method according to claim 9 wherein the step of converting comprises applying to the sequence of representations of relative pitch changes between successive notes of the hummed melody an autocorrelation based pitch detector.

13. A method according to claim 9 wherein the step of converting comprises applying to the sequence of representations of relative pitch changes between successive notes of the hummed melody an autocorrelation based pitch detector having low pass filtering for removing non-vocal frequencies and center-clipping for eliminating unwanted formant structure.

14. A method according to claim 9 wherein the step of converting comprises applying to the sequence of representations of relative pitch changes between successive notes of the hummed melody a wavelet pitch tracker.

15. A method according to claim 9 wherein the step of converting comprises applying to the sequence of representations of relative pitch changes between successive notes of the hummed melody a cubic spline wavelet pitch tracker.

16. A method according to claim 9 further comprising operating the computer means to obtain approximate matches of sequences of digitized representations of relative pitch differences between successive notes.

* * * * *

US006970886B1

(12) **United States Patent**
Conwell et al.

(10) **Patent No.: US 6,970,886 B1**
(45) **Date of Patent: Nov. 29, 2005**

(54) **CONSUMER DRIVEN METHODS FOR ASSOCIATING CONTENT INDENTIFIERS WITH RELATED WEB ADDRESSES**

(75) Inventors: **William Y. Conwell**, Portland, OR (US); **Kenneth L. Levy**, Stevenson, WA (US)

(73) Assignee: **Digimarc Corporation**, Beaverton, OR (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 642 days.

(21) Appl. No.: **09/578,551**

(22) Filed: **May 25, 2000**

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/104.1**; 707/3
(58) **Field of Search** 707/1-10, 100, 707/104.1, 200, 205; 709/223-225, 219, 709/217; 705/10, 14

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,122,403	A	9/2000	Rhoads	
6,269,361	B1 *	7/2001	Davis et al.	707/2
6,286,036	B1	9/2001	Rhoads	
6,311,214	B1	10/2001	Rhoads	
6,324,573	B1	11/2001	Rhoads	
6,389,467	B1 *	5/2002	Eyal	709/223
6,401,118	B1 *	6/2002	Thomas	707/4
2001/0001854	A1	5/2001	Schena et al.	
2001/0020242	A1	9/2001	Gupta et al.	
2001/0055391	A1	12/2001	Jacobs	
2001/0056573	A1	12/2001	Kavac et al.	
2002/0071556	A1	6/2002	Moskowitz et al.	

FOREIGN PATENT DOCUMENTS

WO	WO01/06703	1/2001

OTHER PUBLICATIONS

Ponce, B., "The impact of MP3 and the future of digital entertainment products", WWW Artists consortium, NY, USA, Sep. 1999, pp. 68-70.*

"NSI Commandeers Deadbeat Domains," Wired News, Jun. 26, 2000 3 pp.

"Network Solutions Creates Online Marketplace for Customers' Domain Names," Network Solutions, May 10, 2000, 2 pp.

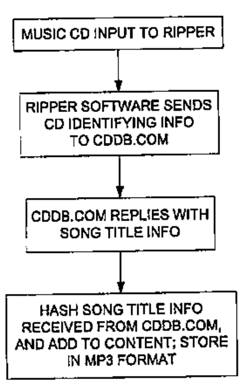
(Continued)

Primary Examiner—Mohammad Ali
(74) *Attorney, Agent, or Firm*—Digimarc Corporation

(57) **ABSTRACT**

Media content objects, such as audio MP3 files, are associated with identifiers. The identifiers can be assigned, or can be implicit (e.g., derived from other data in the content object, as by hashing). A user of the file can utilize the identifier to query a database and thereby obtain the URL of one or more internet resources associated with that content (e.g., web sites with fan info, concert schedules, opportunities to purchase CDs, etc.). Some identifiers may not be associated with URLs in the database. A user who queries the database with such an identifier (e.g., which may be derived from an independently produced MP3) finds that there is not yet an associated URL. In this case, the user may be given the opportunity to lease this virtual address for a predetermined period, with the privilege of specifying a URL for that identifier. Subsequent users who link from this particular MP3 file thereafter are directed to the URL specified by the first user. In some arrangements, the leasing privilege is awarded through a brief auction, triggered by the first user's discovery that the address is not used. Other users who query the database with that identifier during the period of the auction are permitted to bid. When the first lease period expires, the privilege can be re-auctioned. Proceeds from such auctions can be shared, e.g., with the user who triggered the first action, or with the high bidder of a previous auction.

27 Claims, 2 Drawing Sheets



DEPOSITION EXHIBIT
PENGAD 800-631-6989
Karypis 12
11/12/18

Google Ex. 1009

Google Ex. 1020

OTHER PUBLICATIONS

"InSearchofMy.com—Frequently Asked Questions," Network Solutions, 2000, 3 pp., Jul. 29, 2000.
"AddressAuction.com—The World's Only Truly Interactive Market for Secondary Domain names," Copyright 1999, 1 page.
U.S. Appl. No. 09/476,686, filed Dec. 30, 1999, Rhoads et al.
U.S. Appl. No. 09/531,076, filed Mar. 18, 2000, Rhoads et al.
U.S. Appl. No. 09/563,664, filed May 2, 2000, Levy et al.
U.S. Appl. No. 09/574,726, filed May 18, 2000, Rhoads et al.
U.S. Appl. No. 09/547,664, filed Apr. 12, 2000, Rhoads et al.
U.S. Appl. No. 09/571,422, filed May 15, 2000, Rhoads et al.
U.S. Appl. No. 09/292,569, filed Apr. 15, 1999, Rhoads et al.
U.S. Appl. No. 09/502,542, filed Feb. 10, 2000, Davis et al.
U.S. Appl. No. 60/189,246, filed Mar. 14, 2000, Levy.
U.S. Appl. No. 60/134,782, filed May 19, 1999, Rhoads.

* cited by examiner

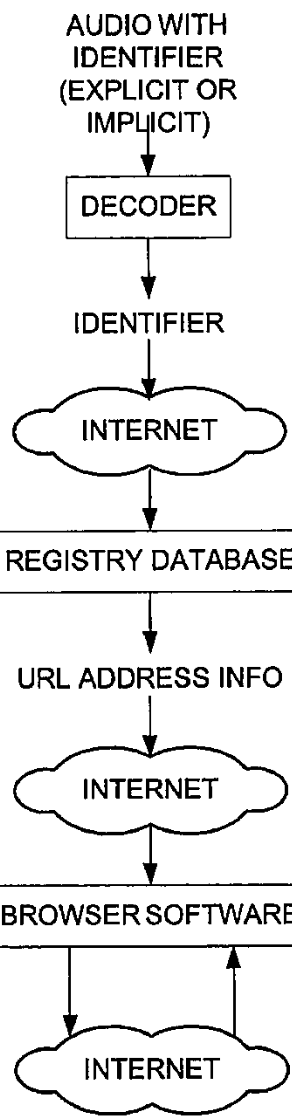

AUDIO WITH IDENTIFIER (EXPLICIT OR IMPLICIT)
DECODER
IDENTIFIER
INTERNET
REGISTRY DATABASE
URL ADDRESS INFO
INTERNET
BROWSER SOFTWARE
INTERNET

FIG. 1

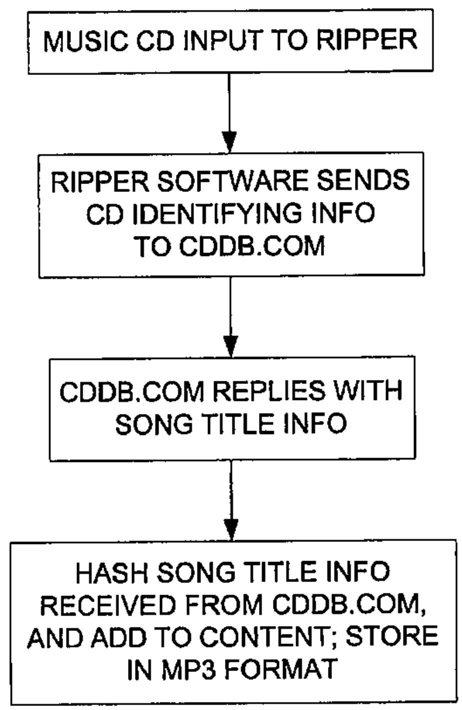

MUSIC CD INPUT TO RIPPER
RIPPER SOFTWARE SENDS CD IDENTIFYING INFO TO CDDB.COM
CDDB.COM REPLIES WITH SONG TITLE INFO
HASH SONG TITLE INFO RECEIVED FROM CDDB.COM, AND ADD TO CONTENT; STORE IN MP3 FORMAT

FIG. 2

034	www.sonymusic.com/catalog/05634.html
112	www.sonymusic.com/catalog/00014.html
198	www.supertracks.com/index/artists/taylor.htm
376	www.emusic.com/0555353x.pdf
597	www.cdw.com/music/featured_CDs/index.html
612	www.sonymusic.com/catalog/00231.html
850	www.polygram.com/franklin/adf_234.htm
921	www.loudeye.com/rap/1999/46755646.html

FIG. 3

034	www.sonymusic.com/catalog/05634.html
112	www.sonymusic.com/catalog/00014.html
198	www.supertracks.com/index/artists/taylor.htm
376	www.emusic.com/0555353x.pdf
597	www.cdw.com/music/featured_CDs/index.html
612	www.sonymusic.com/catalog/00231.html
850	www.polygram.com/franklin/adf_234.htm
883	www.userdefined.com/00004.html
921	www.loudeye.com/rap/1999/46755646.html

FIG. 4

CONSUMER DRIVEN METHODS FOR ASSOCIATING CONTENT INDENTIFIERS WITH RELATED WEB ADDRESSES

RELATED APPLICATION DATA

The subject matter of the present application is related to that disclosed in copending application Ser. No. 09/476,686, filed Dec. 30, 1999; 09/531,076, filed Mar. 18, 2000; Ser. No. 09/563,664, filed May 2, 2000; and 09/574,726, filed May 18, 2000. The disclosures of these applications are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to leasing of virtual addresses, as may be associated with music or other media content.

BACKGROUND AND SUMMARY OF THE INVENTION

For expository convenience, the present disclosure is illustrated with reference to audio content. However, it should be recognized that the principles described below are applicable in any media context, including still imagery, video, product packaging, etc.

In the cited patent applications, the present assignee disclosed a variety of technologies by which audio content can be associated with corresponding internet resources. In some such approaches, the audio content is steganographically encoded (e.g., by digital watermarking) to convey an identifier. When a computer encounters such an encoded audio object, it discerns the encoded identifier, forwards the identifier to a remote database (a "Registry database"), and receives in response—from a database record indexed by the identifier—the address of one or more internet resources related to that audio (e.g., fan sites, concert schedules, e-commerce opportunities, etc.) The computer can then link to such a resource and present same to a user, e.g., using an internet browser program. Such an arrangement is shown in FIG. 1.

There are many variations on this model. For example, instead of steganographically encoding the identifier in the content, the identifier can be added into header or other data with which the content is conventionally packaged.

The identifier can be assigned to the content. Or the identifier can be derived, in some manner, from the content.

In the former, assigned identifier case, an entity such as a music publisher (e.g., Sony) or a music distributor (e.g., emusic.com), selects a number for encoding into the content. The number may be selected from a limited range of numbers (e.g., a range of numbers allocated to that publisher by the proprietor of the Registry database), but the number itself is not inherently related to the content with which it is associated.

In the latter case, the identifier is derived from the content, or from other information associated with the content.

One way to derive an identifier is to employ selected bits of the content, itself, as the identifier. For example, in MP3 audio, where the signal is encoded into frames, the Nth bit of the first 128 frames of a musical work can be assembled together into a 128 bit identifier. Or data present in MP3 headers can be used. In another approach, some or all of the content data is processed by a hashing algorithm to yield a 128 bit identifier corresponding to that content. In both of

these cases, the identifier is implicit in the audio itself. That is, no data needs to be added (e.g., in a header, or by steganographic encoding).

When deriving the identifier from associated information, one can use the table of contents (TOC) of the CD or file allocation table of the DVD. The ID can be embedded within the MP3 file, if it is being ripped from the CD or DVD at the time or ID creation. The method of embedding can be embedded via header, footer or frame bits, or via a watermark.

Some techniques for deriving an identifier may rely on external resources. For example, when "ripping" a song from a commercial music CD into MP3 form, many ripper software programs refer to an on-line disc recognition database, found at www.cddb.com, to obtain the title and length of the song. This "table of contents" information can be used to form the identifier, e.g., by selecting predetermined bits, hashing, etc. In this case, the identifier must generally be added to the audio (i.e., it is explicit, as was the case of the assigned identifiers). Such an arrangement is shown in FIG. 2.

The artisan will recognize that there are an essentially infinite number of algorithms by which such derived identifiers can be generated. (It will be noted that derived identifiers may not be unique. That is, two unrelated audio files may—coincidentally—correspond to the same identifier. But by making the identifier sufficiently long (e.g., 128 bits), such occurrences can be made arbitrarily unlikely.)

When an identifier is assigned to content, the entity doing the assigning (e.g., a record label such as Sony, or a music distributor such as Emusic) can ensure that the Registry database has a record corresponding to that identifier. The database record contains, e.g., one or more URL(s) leading to information relating to the audio content.

A different situation arises when the identifier is derived from content. No master authority ensures that the Registry database has a record corresponding to that identifier. Thus, if a college student rips music from a privately-produced CD into an MP3 file, the identifier derived from that music may not point to an active database record in the Registry database. This can also occur with CDs from major or minor labels that don't register all their identifiers. For example, the Kinks' CDs may not be registered by the label owning rights to their albums because of their fall in popularity. However, a general consumer/business person could register the CD's identifier to sell Kinks' paraphernalia. The consumer/business person does not need to make the same amount of revenue as the record label to make the registration process and maintenance worth his/her time.

If the MP3 file so-produced becomes popular, and is widely spread (e.g., through means such as Napster, Gnutella, etc.) a large potential audience may develop for internet resources related to that MP3. The issue then arises: who manages the Registry database address represented by the corresponding identifier?

The present invention addresses this and related issues.

In accordance with one aspect of the present invention, a user who encounters an unused Registry database record is given an option to manage it, or to participate financially in its exploitation.

For example, in one embodiment, when the Registry database first receives a query corresponding to an un-used identifier, the person initiating the query is given an opportunity to lease that identifier for a predetermined period, such as two months. Upon payment of a nominal fee (e.g., $10), the user can specify a URL that will be stored in the

Registry database in association with that identifier, and to which subsequent users will be directed.

In accordance with another aspect, when a user first queries an un-used identifier in the Registry database, an auction commences, with a nominal opening bid (e.g., $10). The auction continues for a short period, such as a week or a month, allowing other persons who encounter such music early in its distribution life to have a chance at gaining the leasehold rights. At the end of the auction, the winner is granted a lease to that identifier for a predetermined period and can specify the URL with which that identifier is associated.

At the end of the predetermined period, the identifier can be leased for a subsequent term—either for a fixed fee (e.g., a multiple of the fee earlier charged), or through an auction.

It will be recognized that this arrangement has certain similarities to the present system for internet domain name registration. A user can query a whois database maintained by Network Solutions and the like to determine whether a domain name is assigned. If it is not, Network Solutions will offer to assign the domain name for a term of years in exchange for a payment. But the present invention serves different needs and is otherwise different in certain details.

The foregoing and other features and advantages of the present invention will be more readily apparent from the detailed description, which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows an arrangement by which media content is linked to associated internet resources.

FIG. 2 shows a process employing an external resource (here www.cddb.com) to derive an identifier corresponding to audio content.

FIG. 3 shows a Registry database with which one embodiment of the present invention is illustrated.

FIG. 4 shows the Registry database of FIG. 3 after entry of a new record.

DETAILED DESCRIPTION

Referring to FIG. 3, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier from audio content and send the identifier to the database, the database responds by returning the URL corresponding to that identifier back to the user device. The user device then directs an internet browser to that URL. By such arrangements, music (e.g., in MP3 format) can serve as a portal to a web site dedicated to the music artist, a web site giving concert schedules for the artist, a web site offering CDs, etc.

In the FIG. 3 example, if the device decodes the identifier '376' from an MP3 file, and queries the database with this data, the database returns the URL www.emusic.com/0555353x.pdf. The user's web browser is then directed to that URL. (For expository convenience, the identifiers are assumed to be in the range 0–1023. In actual implementations, a much larger range would usually be used.)

The just-described sequence of operations is illustrative and is subject to numerous variations—various of which are detailed in the earlier-cited applications.

Now assume that an up-and-coming band ("The Pinecones") releases a song in MP3 format. No identifier is

affirmatively assigned to the MP3 when it is encoded, but compliant players process the MP3 data to derive an identifier. (An exemplary algorithm may take the first ten frames of MP3 data, and select the 100^{th} data bit from each frame, to yield a ten bit identifier.) The derived identifier may be '883.' A listener of the song forwards this '883' identifier to the Registry database, hoping thereby to be linked to supplemental information about The Pinecones or the song. Instead, the Registry reports (e.g., by a default web page) that there is no further information related to that MP3 (i.e., there is no database record corresponding to identifier '883').

In this case, the Registry can invite the listener to remedy this deficiency and to create a web page that will be associated with that song. In exchange for a small fee, the listener is allowed to specify a URL that will be associated with that identifier for a month. If the user is not equipped to design and host a web page, the Registry can provide simple web page authoring tools and a hosting service permitting the listener to easily create a page on-line.

If the listener accepts this invitation, he makes the payment by various known methods (e.g., via credit card, by one of several emerging web currencies, etc.). He then composes (or specifies) a web page to correspond to that song. The Registry is updated to include a new record for identifier '883,' and includes a link to the page specified by the listener (e.g., www.userdefined.com/00004.html). The table after updating is shown in FIG. 4.

If the user wishes, he can complement the substance of the linked page with banner advertising, co-op links to on-line retailers (e.g., Amazon.com), or other revenue-producing uses.

Other copies of the same bit-rate Pinecones' MP3 file likewise do not have an assigned identifier. However, compliant players will all derive identifiers using the same algorithm, so all other listeners of the song will be directed to the same '883' identifier in the Registry database. Once the first listener activates such a record, later listeners who link to the Registry will be linked to the www.userdefined.com/00004.html web page specified by the first listener.

Since the ID is automatically generated, it may be different for each bit-rate MP3 release as well as for each CD release. Different bit-rate MP3 releases, such as 128 kbps and 96 kbps releases, produce different bits in the resulting MP3 file; thus, the automatically generated identifiers may be different. CD releases refer to different mixes of the music, not copies of the master CD; thus, different CD releases have different bits on the CD and, correspondingly, the automatically generated identifiers may be different for each CD release. In addition, if the MP3 version is ripped by the consumer from the CD, the compliant ripper should embed the identifier generated from the CD into the MP3 file. This structure is necessary because various consumer rippers produce different bits in the MP3 file, since the MP3 format only defines a standard decoder. In all of these cases, the server has two options. One option is to check and make sure that the same song and artist don't already exist. If they do, the new identifier is linked to the same web page. The second option is to allow each identifier, even if from the same song but different MP3 or CD releases, to have an owner.

Of course, by suitably designing the algorithm by which identifiers are derived, non-identical versions of the same basic content may nonetheless correspond to the same identifier. There is extensive published research on such technology, e.g., hashing algorithms by which similar or related, but non-identical, inputs map to the same hash outputs.

In another embodiment, the first listener does not have an absolute right to lease the identifier. Rather, the first listener is given a "first mover" advantage in a brief auction for that identifier. By placing a minimum starting bid (e.g., $10), an auction for the identifier is commenced, and continues for a week. Subsequent listeners who link to that identifier during the auction are given the opportunity to beat the then-highest bid. At the end of the auction period, the high bidder is charged (typically by pre-arranged means), and given the opportunity to specify a link for that identifier. (Again, the Registry operator may offer to host the linked page.)

In a variant of the foregoing, the final bid proceeds are split, with the Registry proprietor sharing a portion (e.g., 5–50%) of the proceeds with the listener who initiated the auction. This may create a strong incentive for use of the system, as listeners try to find music not already linked by the database, hoping to start auctions and share in their proceeds.

Assume the initial lease is for a period of two months. During that period The Pinecones have become wildly popular, and thousands of listeners are linking to the corresponding web page daily. The link is now a hot property. At the expiry of the initial lease term, the Registry proprietor can re-auction the link. The band or its promoters may naturally be one of the bidders. This time the auction may result in large bids, commensurate with the popularity of the music to which it corresponds. Again, the proceeds of the auction may be shared by the Registry proprietor, e.g., with the original listener who discovered the identifier, or with the party who was high bidder in the previous (initial) auction.

In similar fashion, the linking rights can be re-auctioned periodically, with the price being proportional to the music's then-current popularity.

It will be recognized that much of the internet is pornography, and some of the persons leasing identifier links from the Registry may seek to promote pornographic or other inappropriate sites by such links. Accordingly, the Registry may place certain limitations on the linked sites. The sites may be checked for RSAC ratings, and only sites with non-adult ratings may be allowed. Alternatively (or additionally), the sites may be automatically scanned for keywords or content (using intelligent search engines, possibly based upon trained networks and/or artificial intelligence) suggesting pornography, and those having such words may be manually reviewed. Etc.

The proprietor may also require that the linked pages contain at least a threshold amount of non-advertising content (e.g., 50% of screen display)—again to encourage use of the identifiers as links to bonafide resources related to the corresponding audio content.

The problem of automated "bots" querying all possible identifiers in the Registry in an attempt to identify and usurp the inactive entries is mitigated by (a) the huge universe of such identifiers, and (b) the costs of registering.

The maintenance of the table 12 is well understood by those skilled in data structures. For ease of description, the present disclosure assumes that the entries are sorted, by identifier. In actual implementation, this may not be the case. The system may be keyed by identifier, song and artist, thus increasing the speed at which the system can find duplicate songs with different identifiers.

From the foregoing, it will be recognized that embodiments of the present invention can be utilized to spur grassroots development of internet resources associated with a wide variety of media content objects. Commerce in a new class of virtual assets is enabled—offering the possibility of significant financial returns to individuals who have a knack for identifying popular music before it becomes popular. A link that was first leased by a high-schooler for $10 may later be re-leased to Sony Music for $10,000. The virtual real estate represented by these identifiers is priced, over time, commensurately with their changing commercial importance.

Having described and illustrated the principles of the invention with reference to illustrative embodiments, it should be recognized that the invention is not so limited.

For example, while the discussion contemplated that the unassigned identifiers were derived from the content, the same or similar approaches can be applied with assigned identifiers.

Likewise, it will be recognized that the universe of identifiers may be segmented in various ways to achieve various purposes, and only a subset of the entire universe of possible identifiers may be made available in the manners here described. For example, derived identifiers may be designed to map into a lower half of a universe of possible identifiers, with the upper half being reserved for assigned identifiers, i.e. setting the most significant bit to determine whether the identifier is assigned or automatically generated.

The principles described herein are applicable in other contexts and in other applications, e.g., wherever there exists a large universe of identifiers, some of which correspond to objects, and that correspondence is not initially known to an entity controlling usage to which the identifiers are put. To name but one alternative, when video compression and Internet bandwidth improve, this auction system can be applied to videos.

While the explicit identifiers detailed above took the form of watermarks and header data, these are illustrative only; any form of identifier can be similarly treated as virtual real estate and granted to its discover. Thus, systems based on barcodes and other identifiers may make use of the principles of this invention.

To provide a comprehensive disclosure without unduly lengthening this specification, the patents and applications cited herein are incorporated herein by reference.

It should be recognized that the particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.

In view of the wide variety of embodiments to which the principles and features discussed above can be applied, it should be apparent that the detailed embodiments are illustrative only and should not be taken as limiting the scope of the invention. Rather, I claim as my invention all such modifications as may come within the scope and spirit of the following claims and equivalents thereof.

We claim:

1. A method of operating a database that has plural records, the method including receiving queries, each including an identifier, and replying to said queries by reference to information from database records associated with said identifiers, said identifiers being drawn from a universe of possible identifiers, a majority of which do not have active database records associated therewith, the method including:

receiving a query from a user including an identifier that has no active database record associated therewith, said identifier being derived from an existing media content object; and

permitting the user to create an active database record corresponding to said identifier.

2. The method of claim 1 that includes allowing the user to pay a fee, said fee entitling the user to specify at least a portion of the database record corresponding to said identifier.

3. The method of claim 1 that includes allowing the user to make a first bid in an auction, said auction continuing for a predetermined period commencing with said first bid.

4. A method comprising:

deriving an identifier corresponding to an existing media content object;

querying a database with the derived identifier; and

if the database has no active record corresponding to said derived identifier, permitting a party who first queried the database with said identifier to define such a record.

5. The method of claim 4 in which the media content object is an audio file.

6. The method of claim 4 in which the media content object is an MP3 audio file.

7. The method of claim 4 in which the media content object is a video file.

8. The method of claim 4 in which the deriving includes consulting a resource external of the media content object.

9. The method of claim 8 in which the resource is a database.

10. The method of claim 4 in which the deriving includes processing data from the media content object to obtain said identifier.

11. The method of claim 4 in which several identifiers can correspond to the same media content object.

12. The method of claim 11 in which the identifiers are automatically generated from different releases of an audio CD, wherein the releases have different audio and/or table of contents.

13. The method of claim 11 in which the identifiers are automatically generated from different versions of an MP3 file, wherein the versions have different bits due to the compression and/or bit-rate.

14. A method of managing a universe of identifiers, some of said identifiers being active and having internet resources associated therewith, and others of said identifiers being inactive, the method including receiving a query corresponding to an inactive identifier and, in response, initiating a time-limited auction, a winner of said auction being granted the privilege of associating an internet resource with said identifier for at least a predetermined time period.

15. The method of claim 14 in which said active identifiers correspond to different audio content, and the internet resources corresponding to said active identifiers correspond to said audio content.

16. A method comprising:

auctioning to the highest bidder the privilege of defining a link that is to be associated, for a predetermined time period, with an identifier through a database; and

at the expiry of said predetermined time period, re-auctioning said privilege.

17. The method of claim 16 in which the proceeds of said re-auctioning are shared with the high bidder of a previous auction for said privilege.

18. The method of claim 16 in which the identifier corresponds to an existing media content object.

19. The method of claim 18 in which the identifier is derived, rather than assigned.

20. The method of claim 1 wherein a primary function of the database is to link consumers to internet resources, such as web pages, that promote goods or services that are related to the media content objects and that are offered by commercial entities, and said user is one of said consumers, wherein the consumer can participate in such linking in a manner customarily reserved to the commercial entities.

21. The method of claim 1 that includes automatically providing the identifier from a process on a user device—such as a computer—to the database, without requiring the user to type or otherwise manually enter the identifier.

22. The method of claim 4 wherein a primary function of the database is to link consumers to internet resources, such as web pages, that promote goods or services that are related to media content objects and that are offered by commercial entities, and said party is one of said consumers, wherein the consumer can participate in such linking in a manner customarily reserved to the commercial entities.

23. The method of claim 1 that includes automatically providing the identifier from a process on a device maintained by said party—such as a computer—to the database, without requiring said party to type or otherwise manually enter the identifier.

24. The method of claim 14 wherein said identifiers and internet resources are associated through a database, a primary function of which is to link consumers to internet resources that promote goods or services that are related to media content objects and that are offered by commercial entities, and said winner is one of said consumers, wherein the consumer can participate in such linking in a manner customarily reserved to the commercial entities.

25. The method of claim 14 that includes automatically deriving the identifier using a device maintained by said winner, without requiring said winner to type or otherwise manually enter the identifier.

26. The method of claim 16 wherein a primary function of the database is to link consumers to internet resources that promote goods or services that are related to media content objects.

27. The method of claim 16 that includes automatically deriving the identifier from a media content object.

* * * * *