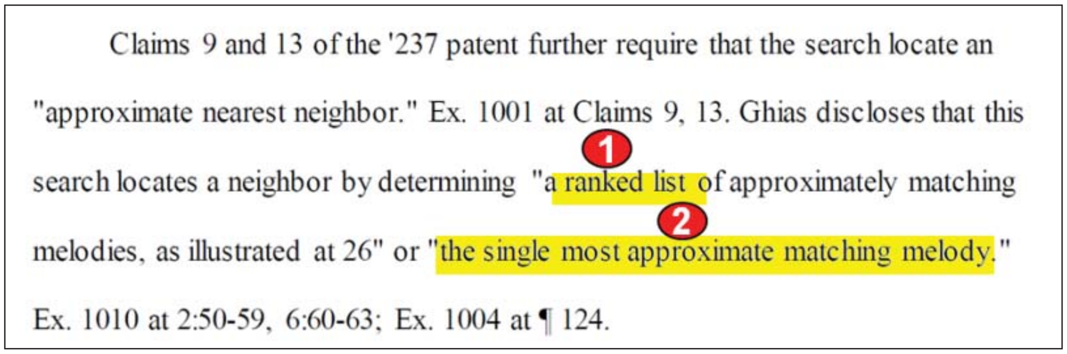
247. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias discloses the claimed "approximate nearest neighbor search." The Petition and corresponding declaration assert that Ghias discloses the "approximate nearest neighbor search" because it produces:

(1) "a ranked list of approximately matching melodies" (labeled ①); or

(2) "the single most approximate matching melody"(labeled②):

248. <u>Petition</u>:

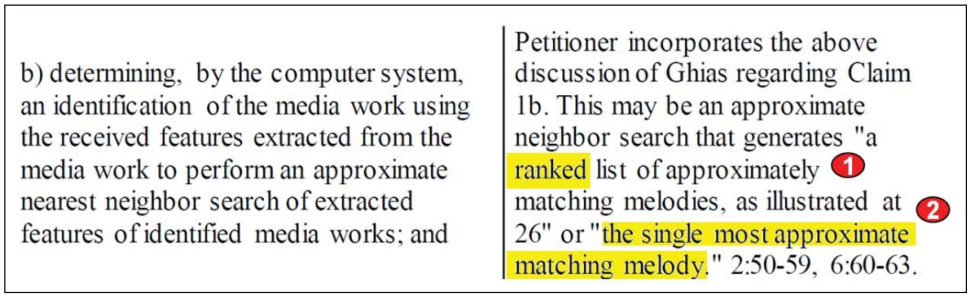


Claims 9 and 13 of the '237 patent further require that the search locate an "approximate nearest neighbor." Ex. 1001 at Claims 9, 13. Ghias discloses that this search locates a neighbor by determining ① "a ranked list of approximately matching ② melodies, as illustrated at 26" or "the single most approximate matching melody." Ex. 1010 at 2:50-59, 6:60-63; Ex. 1004 at ¶ 124.

Pet. ('237) at 42.

249. <u>Petition Charts</u>:

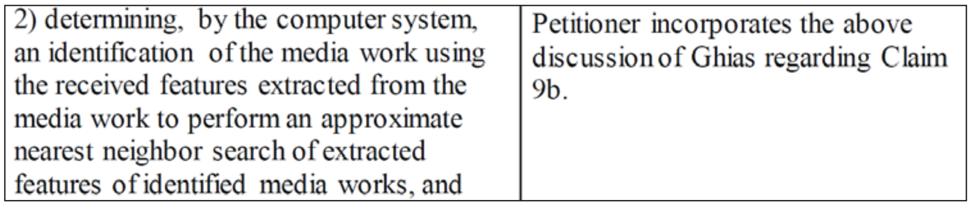
<u>Claim 9(b)</u>:



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	Petitioner incorporates the above discussion of Ghias regarding Claim 1b. This may be an approximate neighbor search that generates "a ranked list of approximately ① matching melodies, as illustrated at ② 26" or "the single most approximate matching melody." 2:50-59, 6:60-63.

Pet. ('237) at 45.

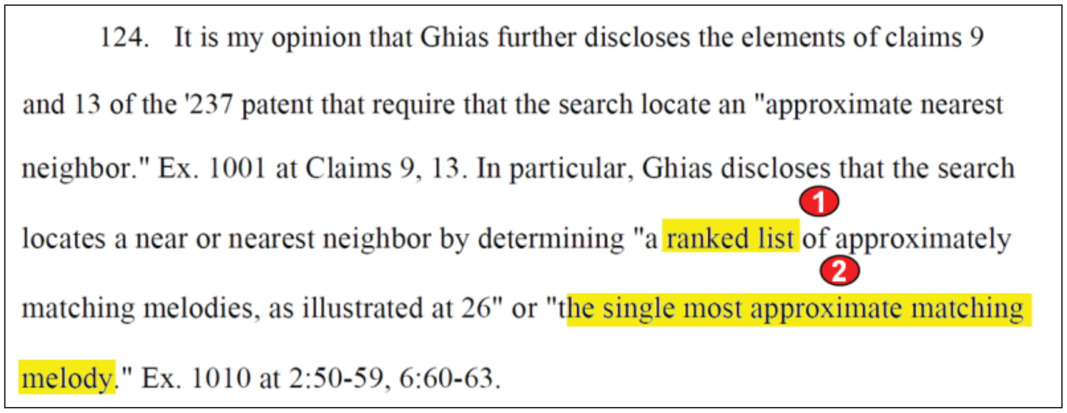
<u>Claim 13(b.2) (referencing claim element 9(b))</u>:



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	Petitioner incorporates the above discussion of Ghias regarding Claim 9b.

Pet. at 46.

250. <u>Declaration</u>:

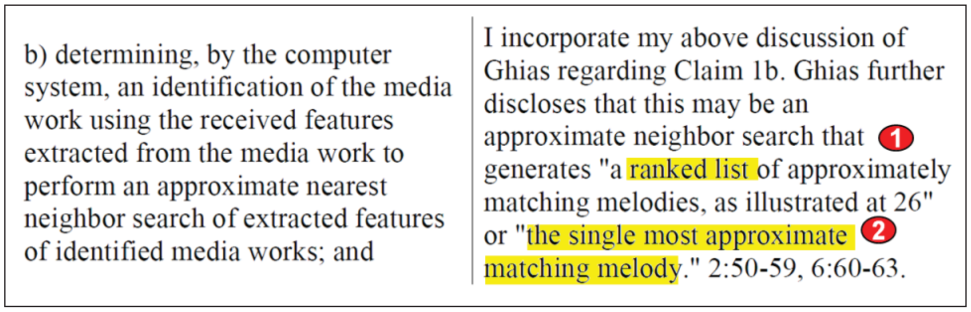


124. It is my opinion that Ghias further discloses the elements of claims 9 and 13 of the '237 patent that require that the search locate an "approximate nearest neighbor." Ex. 1001 at Claims 9, 13. In particular, Ghias discloses that the search locates a near or nearest neighbor by determining "a ranked list ① of approximately matching melodies, as illustrated at 26" or "the single most approximate matching ② melody." Ex. 1010 at 2:50-59, 6:60-63.

Moulin Decl. ('237) ¶124.

251. <u>Declaration Charts</u>:

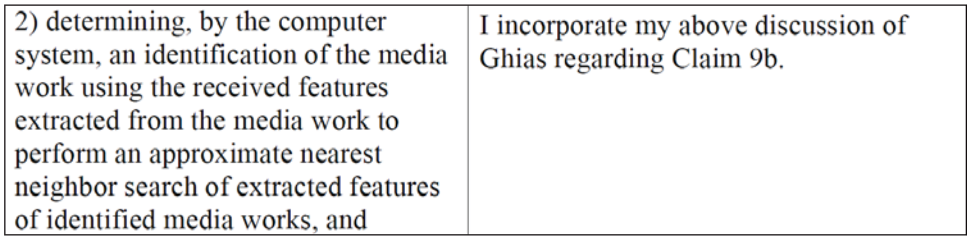
<u>Claim 9(b)</u>:



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	I incorporate my above discussion of Ghias regarding Claim 1b. Ghias further discloses that this may be an approximate neighbor search that ① generates "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate ② matching melody." 2:50-59, 6:60-63.

Moulin Decl. ('237) ¶127.

<u>Claim 13(b.2) referencing claim element 9(b))</u>:

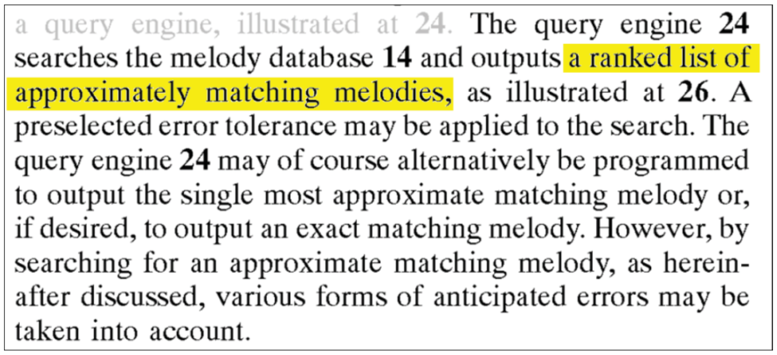


2) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform an approximate nearest neighbor search of extracted features of identified media works, and	I incorporate my above discussion of Ghias regarding Claim 9b.

Moulin Decl. (‘237) ¶127.

252. One skilled in the art would understand that neither of the cited passages discloses the claimed “approximate nearest neighbor search” because, as described above, both the ranked list and single most approximate matching melody always identify the closest match. I address each passage in turn:

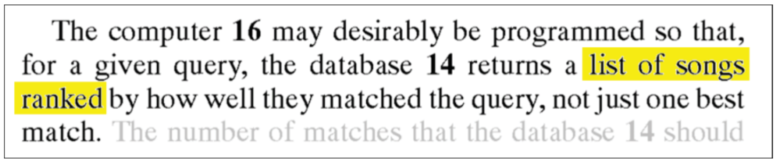
253. Passage 1:



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.

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 “approximate nearest neighbor search.” An “approximate nearest neighbor search” must identify “a close, but not necessarily exact or closest, match” Section V(D); Decision (‘237) at 8. Both outputs disclosed in this passage necessarily disclose an exact or the closest match and, therefore, are not an "approximate nearest neighbor search."

254. Passage 2:

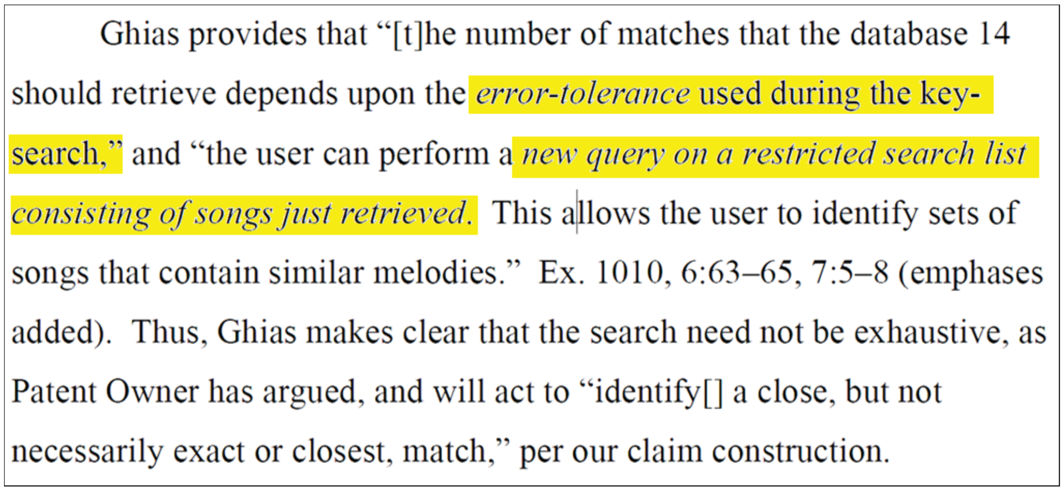

The computer **16** may desirably be programmed so that, for a given query, the database **14** returns a list of songs ranked by how well they matched the query, not just one best match. The number of matches that the database **14** should

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

255. Moreover, under the proper construction of "approximate nearest neighbor search," the search must be a sub-linear search. '237, 9:12-19 (an approximate nearest neighbor search is an "example of a sub-linear time search"); Section V(D). As demonstrated above, these passages disclose a linear (rather than sublinear) search.

256. *Board's concerns*: I now address the Board's specific concerns (identified in its Decision) with respect to whether Ghias discloses the claimed "approximate nearest neighbor search." In instituting Ground 2 of the '237 IPR, the Board found that Ghias disclosed the "approximate nearest neighbor search"

because the error-tolerance search disclosed in Ghias "allows the user to identify sets of songs that contain similar melodies:"



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 ('237) at 18-19. The Board did not explain, however, how "Ghias makes clear" that the search in Ghias will "identify[] a close, but not necessarily exact or closest, match" as required by an "approximate nearest neighbor search."

257. The Board noted that using an "error-tolerance," the user can adjust the number of output matches ("The number of matches that the database 14 should retrieve depends upon the error-tolerance used during the key search." Ghias, 6:63-65); and a new query can be performed on the restricted list ("If the list is too large, the user can perform a new query on a restricted search list consisting of songs just retrieved." Ghias, 7:5-8). But nothing in these passages or anywhere else in Ghias states or even suggests that the output of the initial list or the output of the restricted search will "identify a close, but not necessarily exact or closest, match." As I explained above, no such search is expressed in Ghias or is

inherent (*i.e.*, necessarily present). Rather, the search will always ("necessarily") identify an exact or closest match. Accordingly, Ghias does not disclose the claimed "approximate nearest neighbor search."[30]

C. '237 Ground 3: The instituted claims of the '237 patent are not obvious over Iwamura and Chen.

258. 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, those references would still be missing an important element and therefore, even with the combination, one of ordinary skill would still not possess the invention.

259. Any combination of Iwamura with Chen would still be missing the same elements addressed above in Ground 1.

260. Ground 3 is directed to only dependent claims 26 and 27 which depend either directly or indirectly on independent claim 25; and claims 34 and 35 which depend either directly or indirectly on independent claim 33. Pet. ('237) at

[30] An approximate nearest neighbor search could miss one or more of the closest matches in the returned search results. The searches disclosed in Ghias never purport to miss one or more of the closest matches in the returned results.

53-56; Decision (‘237) at 22. Ground 3 presents two alternative grounds—that the dependent claims “are obvious over Iwamura alone, or alternatively, over Iwamura in view of Chen.” Pet. (‘237) at 53.

261. As I demonstrated above, Iwamura does not disclose key elements from the independent claims upon which Ground 3 is based (claims 25 and 33) including:

- “non-exhaustive search … to identify a near neighbor” (claim 25(b.2); and
- “approximate nearest neighbor search” (claim 33(b.2)).

I note that Petitioner does not rely on Chen for these elements. Pet. (‘237) at 53-56; Moulin Depo. 371:17-20 (addressing sublinear); Moulin Depo. 372:2-4 (addressing non-exhaustive); Moulin Depo. 372:5-7 (addressing approximate nearest neighbor search).

262. Moreover, I note that Petitioner does not assert that these missing elements are obvious in light of Iwamura but rather continues to assert that they are expressly disclosed in Iwamura. *See e.g.*, Pet. (‘237) 54 (“For the reasons expressed in Ground 1 [anticipation based on Iwamura], Iwamura discloses all elements of claims 25 and 33.”). Accordingly, Ground 3 fails at least because the elements from the independent claims addressed above are missing from Iwamura and the Petition does not identify any basis for correcting these deficiencies based on either Iwamura or Chen.

VII. ‘988 patent.

263. The Board instituted the ‘988 IPR based on three Grounds:

- Ground 1: Claims 15–17, 21–23, 28, 31, and 51 under 35 U.S.C. § 102(b) as anticipated by Ghias;
- Ground 2: Claims 22, 24–26, and 52 under 35 U.S.C. § 103(a) as obvious over Ghias; and
- Ground 3: Claims 15–17, 21, 23, 27, 28, 31–33, 38, and 51 under 35 U.S.C. § 102(e) as anticipated by Iwamura;

Decision (‘988) at 22. I note that the only instituted independent claim is claim 15. I address each Ground in turn.

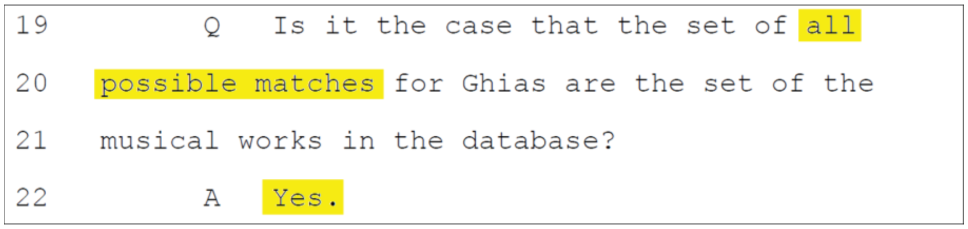
A. ‘998 Ground 1: The instituted claims of the ‘988 Patent are not anticipated by Ghias.

264. The single independent claim of the ‘988 patent instituted for trial requires a “non-exhaustive search identifying a neighbor.” ‘988, claim 15. Ghias does not disclose (1) a non-exhaustive search, (2) a search identifying a neighbor, or (3) determining an action based on the identification. I address each deficiency in turn.

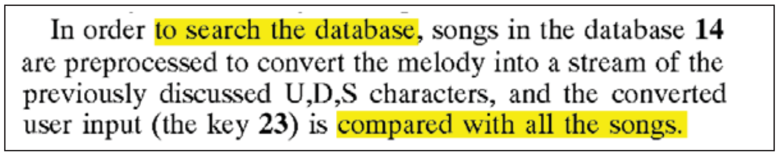
1. non-exhaustive search (claim element 15(b)).

265. 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."

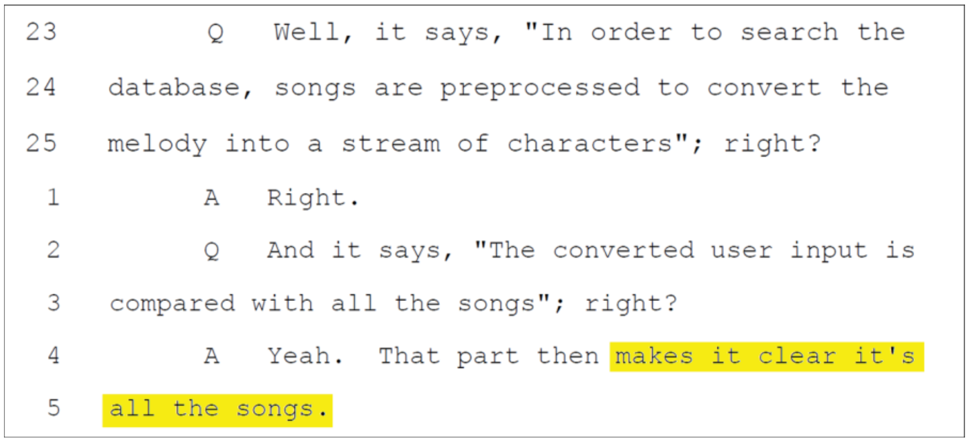
266. One skilled in the art would understand that Ghias teaches an exhaustive search that compares the work to be identified (user input 23) with "all the songs" in the database—*i.e.*, "all possible matches." One skilled in the art would understand that all "possible matches" in the system disclosed in Ghias are all of the songs in the database. My understanding is confirmed by Petitioner's Declarant:


Q Is it the case that the set of all possible matches for Ghias are the set of the musical works in the database?
A Yes.

Moulin Depo. 325:19-22. Ghias discloses a search that compares the work to be identified ("user input") with all possible matches—"all the songs" in the database:


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.

Ghias, 5:66-6:2. As Petitioner's Declarant acknowledged when addressing the paragraph from Ghias quoted above:

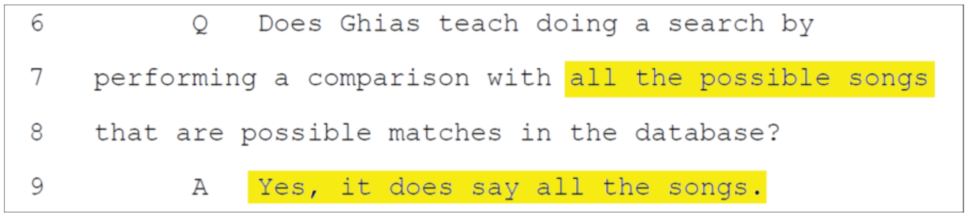

Q Well, it says, "In order to search the database, songs are preprocessed to convert the melody into a stream of characters"; right?

A Right.

Q And it says, "The converted user input is compared with all the songs"; right?

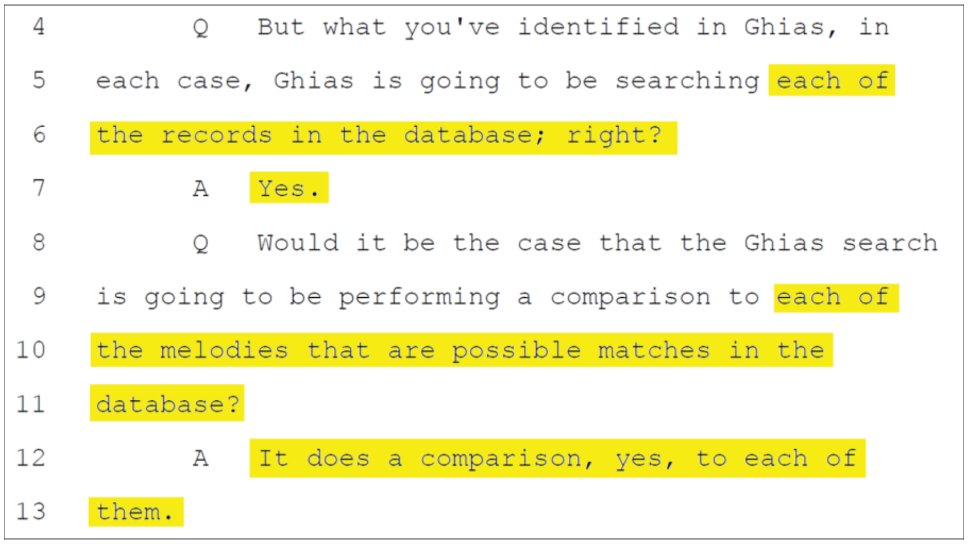
A Yeah. That part then makes it clear it's all the songs.

Moulin Depo. 339:23:-340:5.


Q Does Ghias teach doing a search by performing a comparison with all the possible songs that are possible matches in the database?

A Yes, it does say all the songs.

Moulin Depo. 340:6-9.


Q But what you've identified in Ghias, in each case, Ghias is going to be searching each of the records in the database; right?

A Yes.

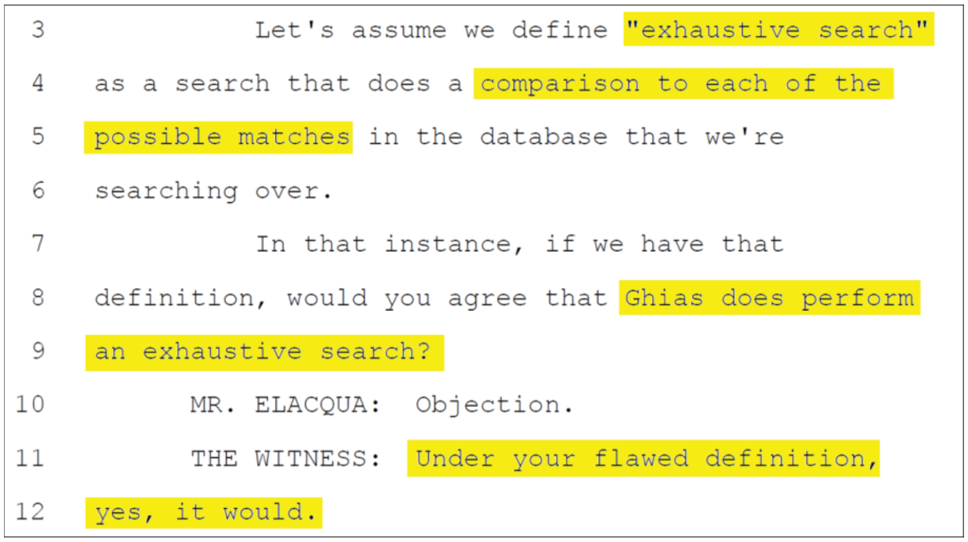
Q Would it be the case that the Ghias search is going to be performing a comparison to each of the melodies that are possible matches in the database?

A It does a comparison, yes, to each of them.

Moulin Depo. 323:4-13.

267. The user input (23) is not compared with some songs in the melody database (14); rather, it "is compared with all the songs." Ghias does not disclose a

search algorithm that does not compare the query to every record in the reference data set. Petitioner's Declarant confirmed my understanding—that the search disclosed in Ghias compares the song to be identified with each record in the database and is therefore not "non-exhaustive"—"a search that locates a match without a comparison of all possible matches" (Section V(B)); Decision ('998) at 7):

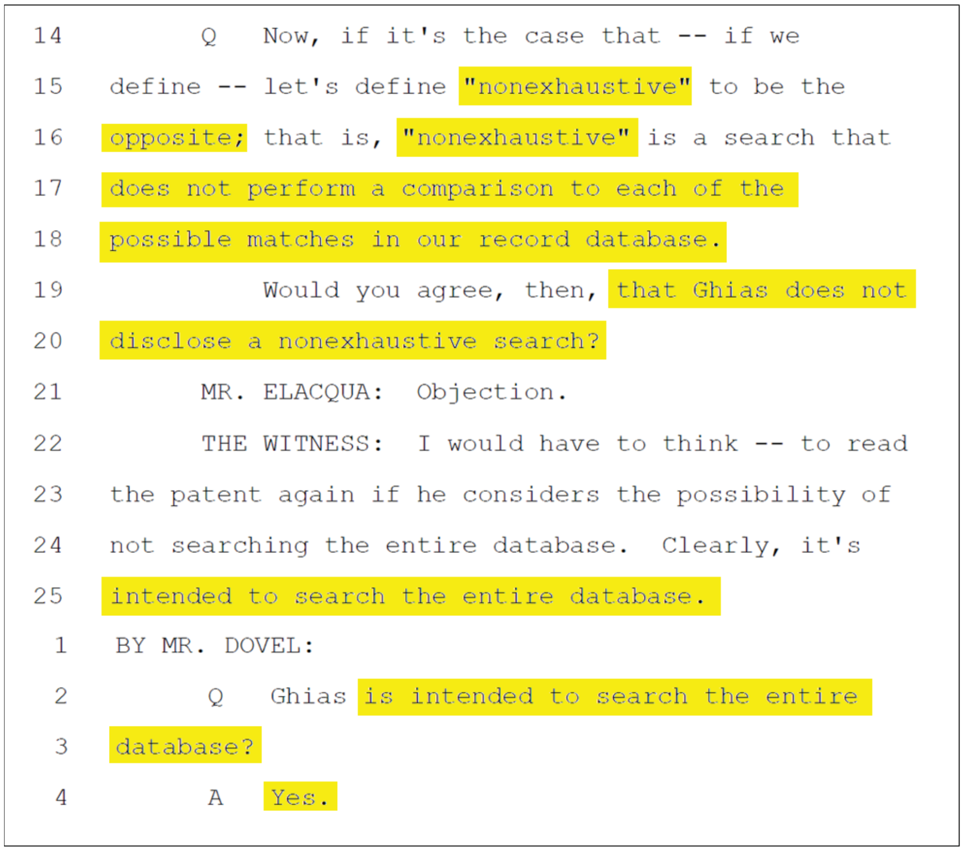

Let's assume we define "exhaustive search" as a search that does a comparison to each of the possible matches in the database that we're searching over.

In that instance, if we have that definition, would you agree that Ghias does perform an exhaustive search?

MR. ELACQUA: Objection.

THE WITNESS: Under your flawed definition, yes, it would.

Moulin Depo. 327:3-12.



Q Now, if it's the case that -- if we define -- let's define "nonexhaustive" to be the opposite; that is, "nonexhaustive" is a search that does not perform a comparison to each of the possible matches in our record database.

Would you agree, then, that Ghias does not disclose a nonexhaustive search?

MR. ELACQUA: Objection.

THE WITNESS: I would have to think -- to read the patent again if he considers the possibility of not searching the entire database. Clearly, it's intended to search the entire database.

BY MR. DOVEL:

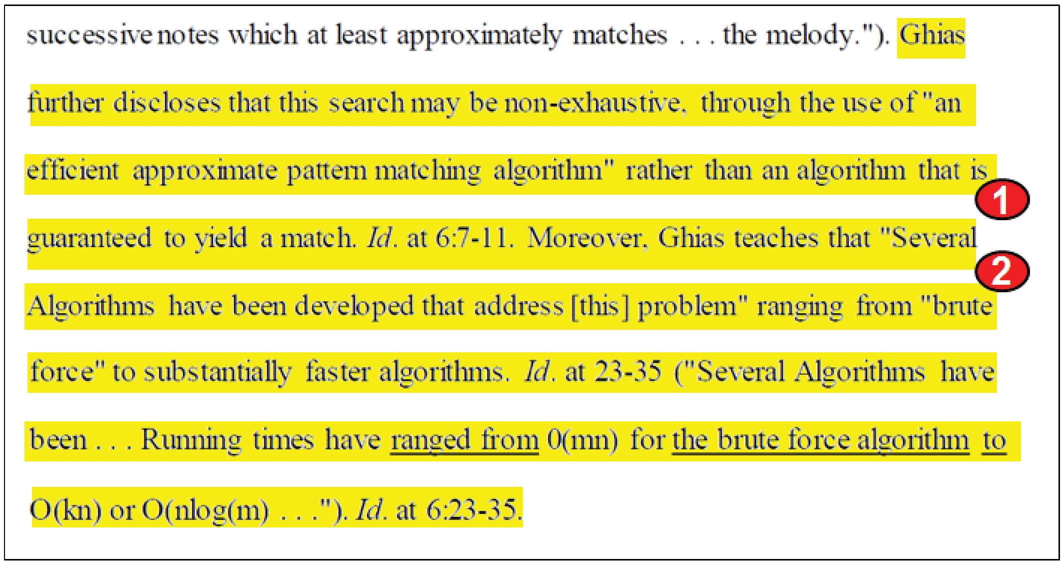
Q Ghias is intended to search the entire database?

A Yes.

Moulin Depo. 327:14-328:4.

268. The Petition and corresponding Declaration fail to demonstrate that Ghias discloses a non-exhaustive search.

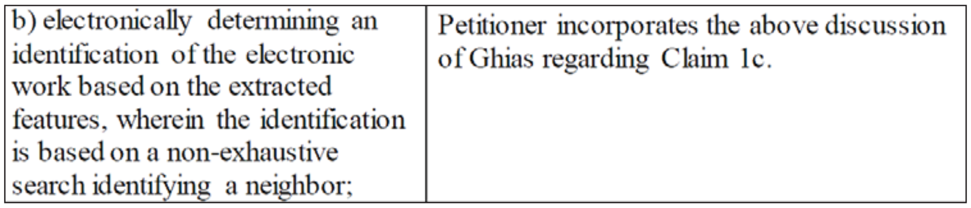
269. <u>Petition</u>: As support for the claimed “non-exhaustive search,” the Petition relies on the following assertions (and corresponding references to Ghias) labeled ❶ and ❷:



successive notes which at least approximately matches . . . the melody."). Ghias further discloses that this search may be non-exhaustive, through the use of "an efficient approximate pattern matching algorithm" rather than an algorithm that is guaranteed to yield a match. *Id.* 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 23-35 ("Several Algorithms have been . . . Running times have ranged from 0(mn) for the brute force algorithm to O(kn) or O(nlog(m) . . ."). *Id.* at 6:23-35.

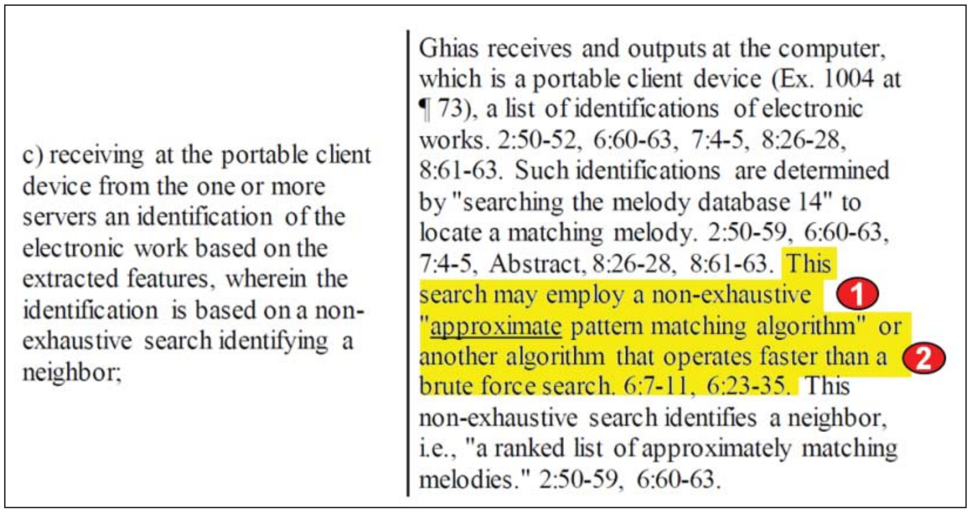
Pet. ('988) 9-10.

270. Petition Charts: The charts in the Petition rely on 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.

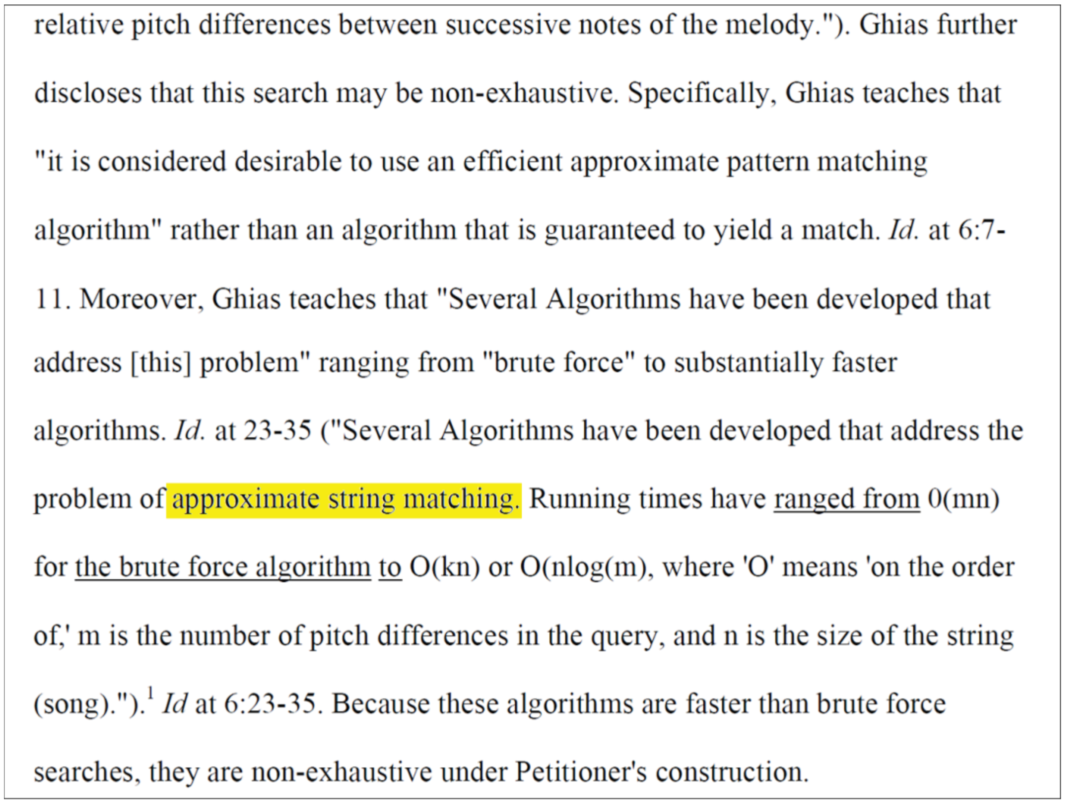
Pet. ('988) at 14. The chart for claim 1, element [c], in turn, provides:



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 ❶ "approximate pattern matching algorithm" or another algorithm that operates faster than a ❷ 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.

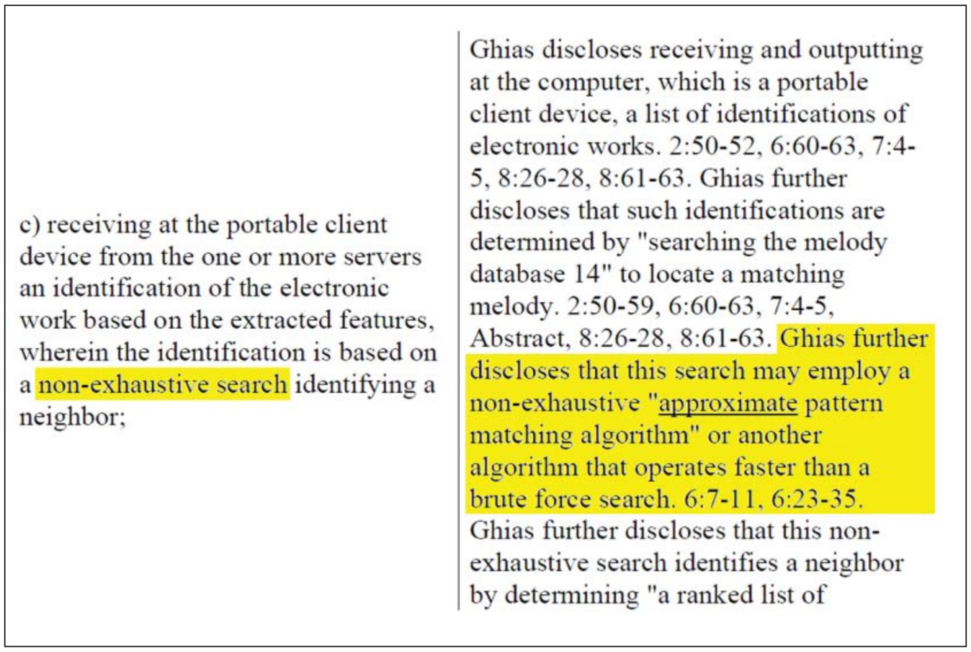
271. Declaration: Petitioner's Declaration relies on the same assertions and passages from Ghias:



relative pitch differences between successive notes of the melody."). 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. *Id.* 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 23-35 ("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 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).").[1] *Id* at 6:23-35. Because these algorithms are faster than brute force searches, they are non-exhaustive under Petitioner's construction.

Moulin Decl. ('988) ¶¶69-70.

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

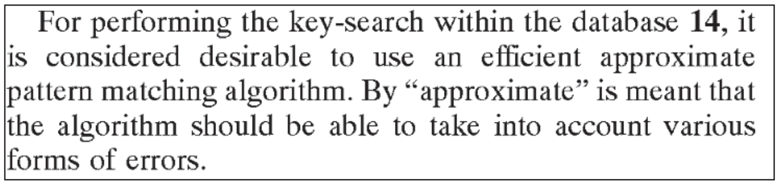


c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;	Ghias discloses receiving and outputting at the computer, which is a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

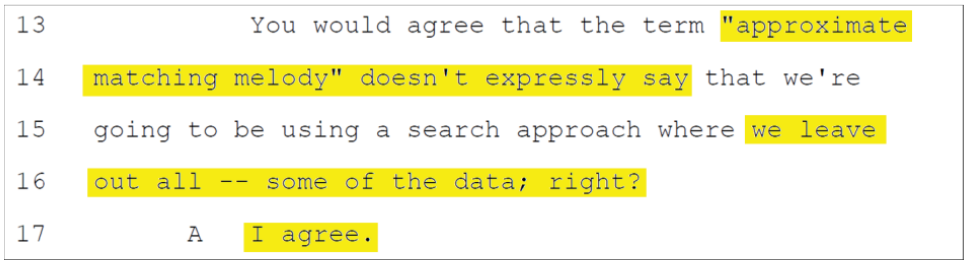
Moulin Decl. (‘988) ¶75.

273. These are the only passages from Ghias cited by the Petitioner and Declarant to support the sub-linear claim elements. Moulin Depo. 113:15-21. The assertions relating to these passages fails to: (a) apply Petitioner’s construction (or any other construction) of non-exhaustive to Ghias; or (b) explain how an “approximate string matching algorithm” is expressly or inherently a non-exhaustive search. One skilled in the art would understand that neither the assertions nor the passages from Ghias disclose the claimed non-exhaustive search. I address each in turn.

274. Passage 1:

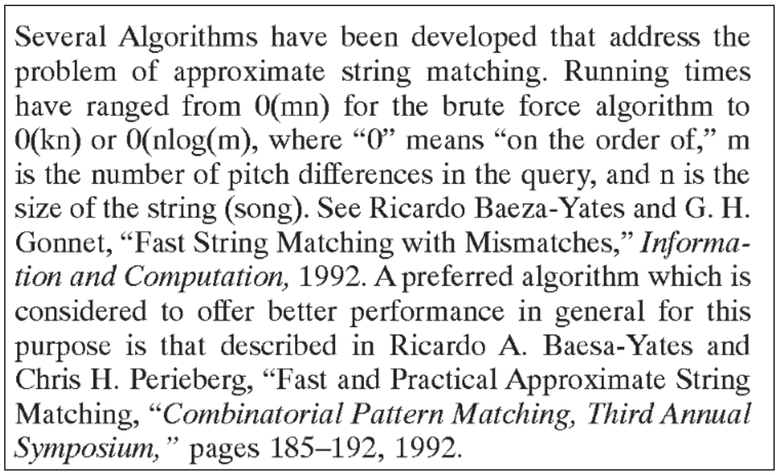

For performing the key-search within the database **14**, it is considered desirable to use an efficient approximate pattern matching algorithm. By "approximate" is meant that the algorithm should be able to take into account various forms of errors.

Ghias, 6:7-11. First, this passage does not state that the algorithm is not guaranteed to yield a match (as interpreted by Petitioner). Second, and more importantly, the described algorithm does not state (or even suggest) that all possible matches in the database are not searched. The passage does not state that all matches are not considered, or even that all data in all possible matches is not considered. My understanding is confirmed by Petitioner's Declarant:


```
          You would agree that the term "approximate
matching melody" doesn't expressly say that we're
going to be using a search approach where we leave
out all -- some of the data; right?
     A    I agree.
```

Moulin Depo. 347:13-17.

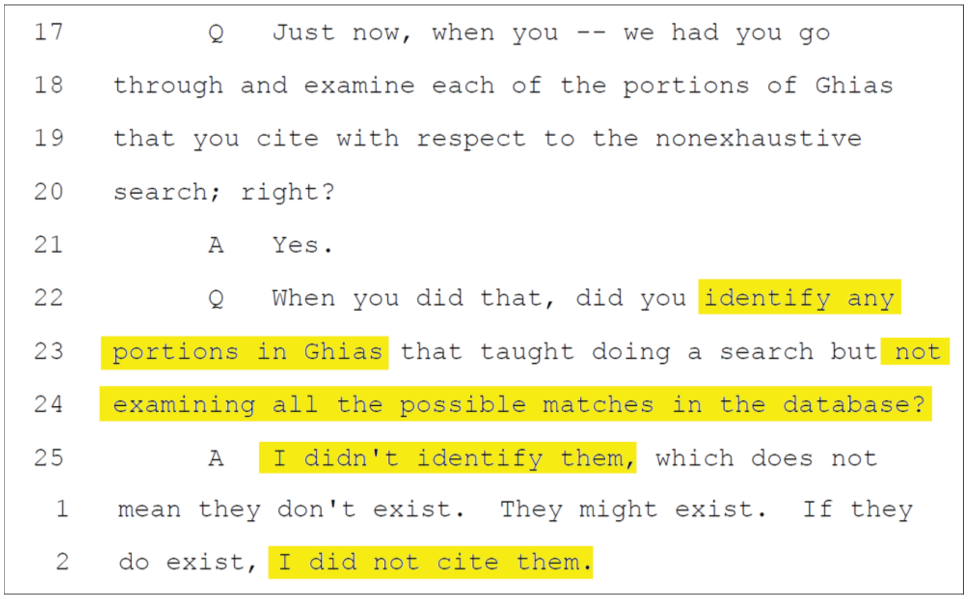
275. <u>Passage 2</u>:


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.

Ghias, 6:23-35. One skilled in the art would understand that the "approximate string matching" algorithms discussed in this passages involve matching a work with a record in the database, where the work to be identified includes an "error" so that "various forms of errors" would not prevent a proper match from being identified. The "approximate string matching" algorithm is applied when the work melody "is compared with all the songs" in the database and all of the data within each record. Ghias, 5:66-6:2; Moulin Depo. 347:13-17. This passage discusses comparing the work with a single record in the database.

276. Accordingly, Ghias does not disclose a search that would even meet Petitioner's improper construction of "non-exhaustive search," because Ghias does not search less than "all possible matches" or even less than "all data within all possible matches."

277. I observed that Petitioner only cited the two passages quoted above as support that Ghias discloses the claimed non-exhaustive search. Petitioner's expert confirmed my observation:



Q Just now, when you -- we had you go through and examine each of the portions of Ghias that you cite with respect to the nonexhaustive search; right?

A Yes.

Q When you did that, did you identify any portions in Ghias that taught doing a search but not examining all the possible matches in the database?

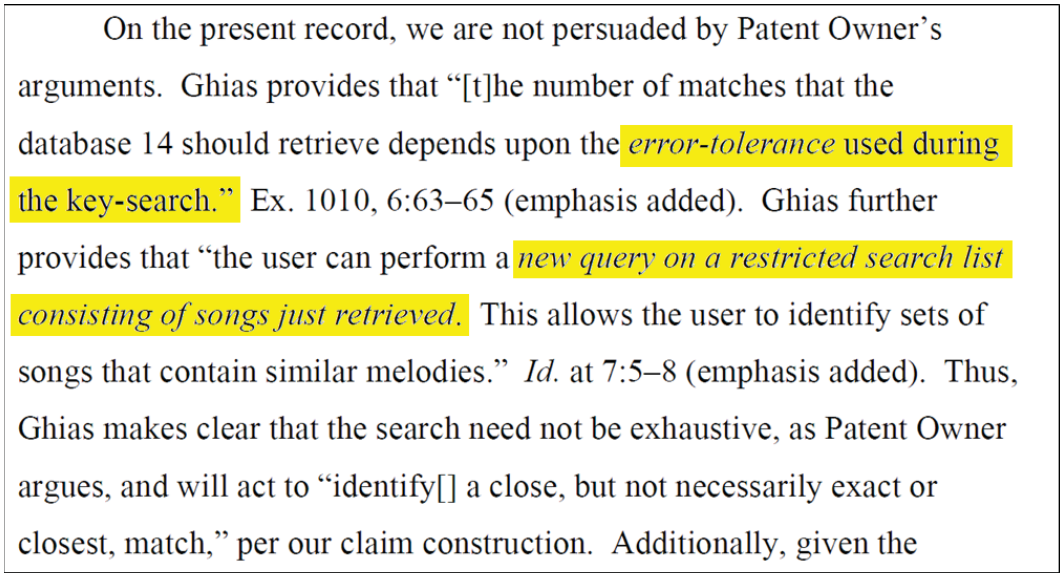
A I didn't identify them, which does not mean they don't exist. They might exist. If they do exist, I did not cite them.

Moulin Depo. 332:17-333:2. As I demonstrated above, these two passages fail to disclose the claimed non-exhaustive search. Accordingly, Petitioner failed to satisfy its burden of establishing that Ghias discloses the claimed non-exhaustive search.

278. Moreover, Petitioner's expert confirmed that other passages from Ghias cited in his Declaration—in an attempt to establish other claimed elements—also do not establish the claimed non-exhaustive search. Moulin Depo. 330:19-331:24; 239:22-25 (2:50-52 "does not teach excluding a portion of the database from our search"); Moulin Depo. 330:15-18 (2:50-52 ("Q. Does Ghias have any portion in where it teaches affirmatively searching only part of the database. A. Not in that sentence, no."); Moulin Depo. 330:1-14 (2:50-52 (the "natural inference" from the statement that the "query engine 24 searches the melody database 14" is that "it's going to search the entire database"); Moulin Depo.

334:2-21 (Q. "[D]oes Ghias teach looking at only a portion of the database? ... A. It does not do that in this paragraph."); Moulin Depo. 337:7-338:17.

279. *Board's concerns*: I now address the Board's specific concerns (identified in its Decision in the '988 IPR) with respect to whether Ghias discloses the claimed non-exhaustive search. I note that 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 in an attempt to establish the claimed non-exhaustive search. Instead, the Board initially 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:"


On the present record, we are not persuaded by Patent Owner's arguments. 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. Additionally, given the

Decision (‘988) at 12. 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 are unpatentable based on Ghias.

280. First, had the concept of a new second search based on the restricted list (and these passages from Ghias cited by the Board) disclosed the claimed “non-exhaustive search” (as I demonstrated below, they do not), it is my understanding that it could be improper for the Board to rely on these passages in finding the challenged claims unpatentable because these passages were not identified by the Petitioner as support for the non-exhaustive search.

281. I note that Petitioner never asserted (in the Petition, charts, or Declaration) that Ghias discloses a non-exhaustive search because the “user can perform a new query on a restricted search list consisting of songs.” The Petition does not even mention the words or concepts emphasized by the Board in its Decision and that form the basis for the Board’s preliminary finding that Ghias discloses a non-exhaustive search: “error-tolerance” and “restricted search list consisting of songs just receive.” The only references to Ghias presented by the Petitioner for the claimed non-exhaustive search are Ghias, 6:7-11 and 6:23-35 addressing approximate string matching, not performing a “new query on a restricted search list consisting of songs just retrieved” based on an error tolerance. Petitioner’s Declarant did not “cite anything in [his] Declaration that teaches, in

Ghias, performing a search that returns a list of ranked matching songs and then performing a second search on that list." Moulin Depo. 146:21-147:21.

282. I note that the Board, however, relied exclusively on two completely different passages from Ghias not cited by Petitioner—Ghias, 6:63-65 and 7:5-8. Decision ('988) at 12. One skilled in the art would understand that these passages address a different concept than the approximate pattern matching concept identified by Petitioner as support for the nonexhaustive search element.

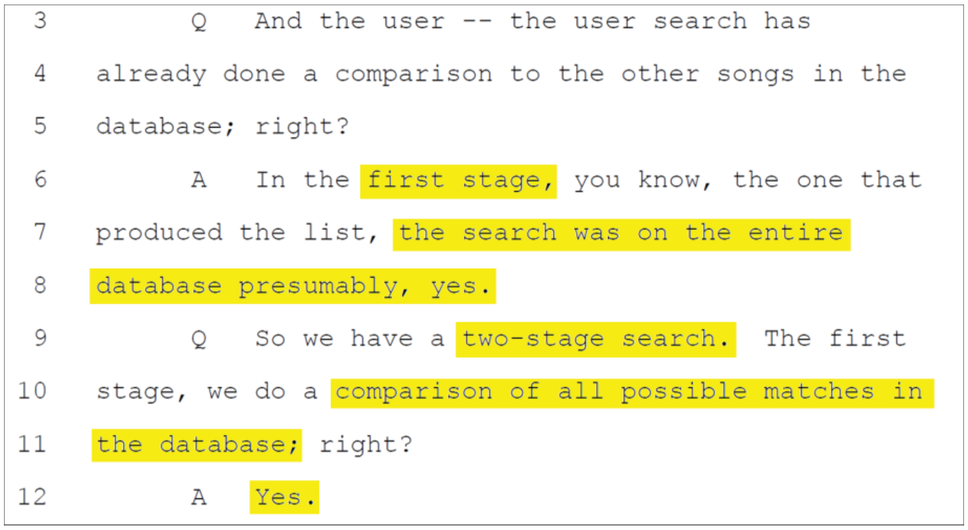
283. Second, using a "new query" on the "restricted search list consisting of songs just received" does not disclose the claimed "non-exhaustive search." A "non-exhaustive search" is "a search that locates a match without a comparison of all possible matches." *See* Section V(B), ¶¶X. The restricted search can be viewed in one of two ways.[31] Under either view, Ghias does not disclose a non-exhaustive search. I address each view in turn.

284. _*First view*_: Under the first view, the search to identify the record that matches the song being hummed is viewed as a single search with two stages. Under this view, the second search on the "restricted list" is not an independent search—the two stages of the search are not independent. Rather, the search on the

[31] Ghias provides no details or information about the search on the restricted search list.

"restricted list" is the second stage of a two-stage search, dependent on the first stage. *See* Moulin Depo. 336:9-15. The second search depends on the first to generate a candidate set. A single work, *i.e.*, the song being hummed (not two or more works), is being identified in the two-stage search. The two stages refine the identified matches; the second stage does not identify any new matches.

285. To constitute a "non-exhaustive search" under this view, the two-stage search process disclosed in Ghias would have to conduct the search without comparing the work to be identified with all possible matches in the dataset. One skilled in the art would understand that the two-stage search disclosed in Ghias is exhaustive because the first stage compares the query to all possible matches in the dataset —"all the songs." Ghias, 5:66-6:2. My understanding is confirmed by Petitioner's Declarant:

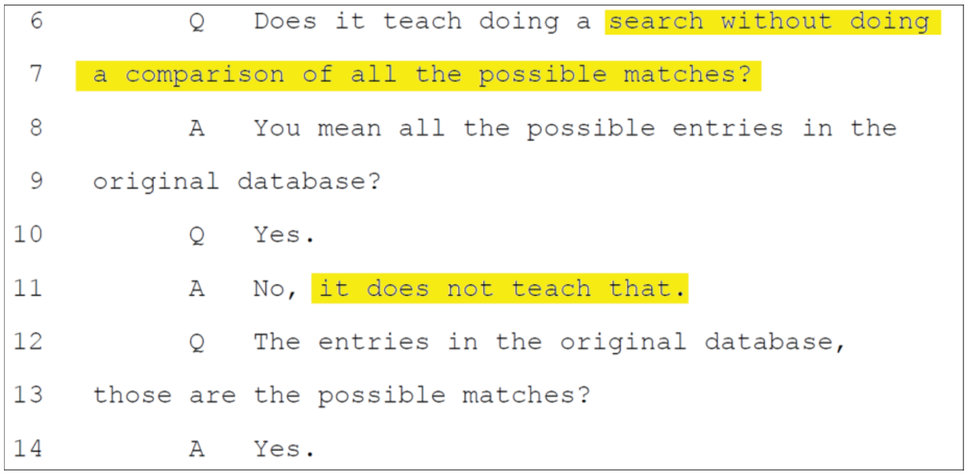

Q And the user -- the user search has already done a comparison to the other songs in the database; right?

A In the first stage, you know, the one that produced the list, the search was on the entire database presumably, yes.

Q So we have a two-stage search. The first stage, we do a comparison of all possible matches in the database; right?

A Yes.

Moulin 336:3-336:12.



Q Does it teach doing a search without doing a comparison of all the possible matches?

A You mean all the possible entries in the original database?

Q Yes.

A No, it does not teach that.

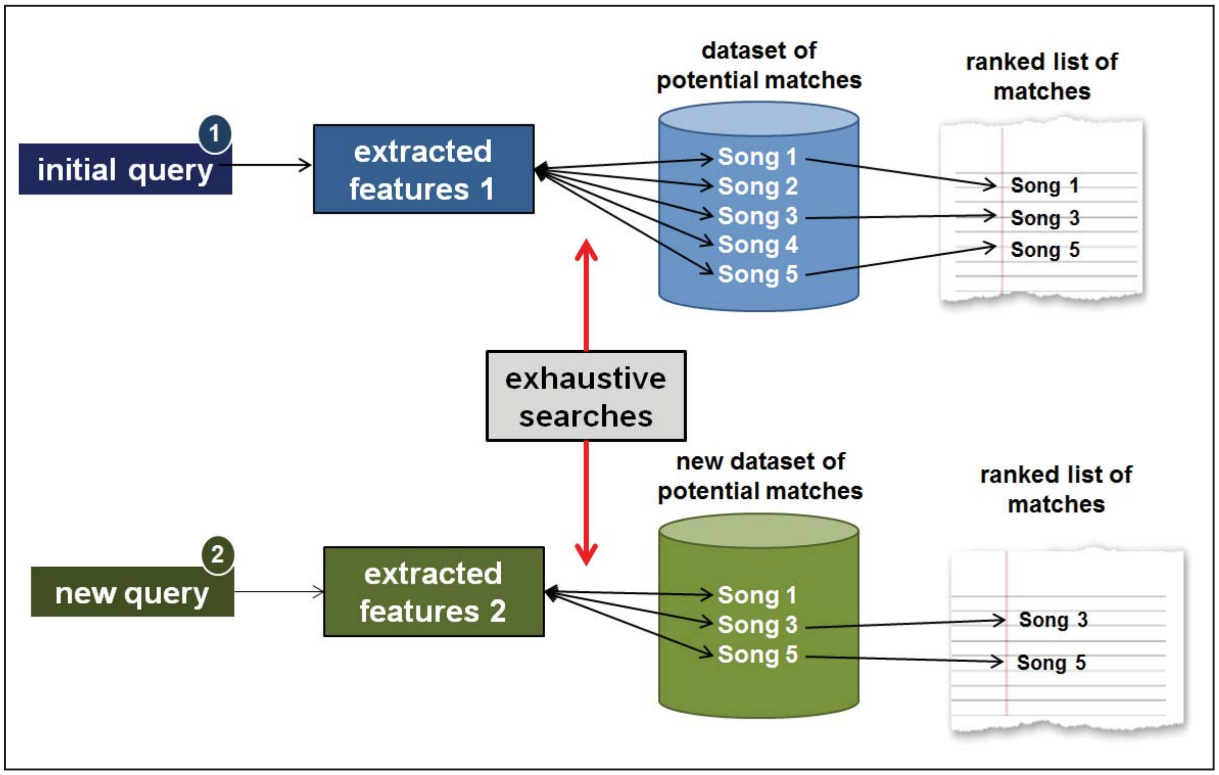
Q The entries in the original database, those are the possible matches?

A Yes.

Moulin Depo. 338:6-14.

286. Under this view, the query on a restricted search list is part of a broader search of every record in the database, which compares the work to be identified with all possible matches—all records in the data set.

287. *Second view*: Alternatively, the second search could be viewed as an independent second search. As disclosed in Ghias, the second search is based on a "new query"—"the user can perform a new query on a restricted search list." Ghias, 7:4-8. The two searches, the first based on an initial query, and the second, based on a second "new query," is reflected in this illustration:



To refine the list of potential matches, the "new query" (2) disclosed in Ghias must be different from the original query (1). This is because Ghias does not teach an alternative search algorithm for searching the restricted list. Rather, Ghias teaches that the same search algorithm is applied to the "new query" (2) that was previously applied to the initial query (1). If the initial query (1) is applied to the restricted list using the same algorithm, the search would produce the same restricted list rather than refine the search as intended by Ghias.[32] Although the

[32] If the query remains constant—the query is not changed—but a different algorithm is applied to the restricted list, this would constitute a single search with

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.

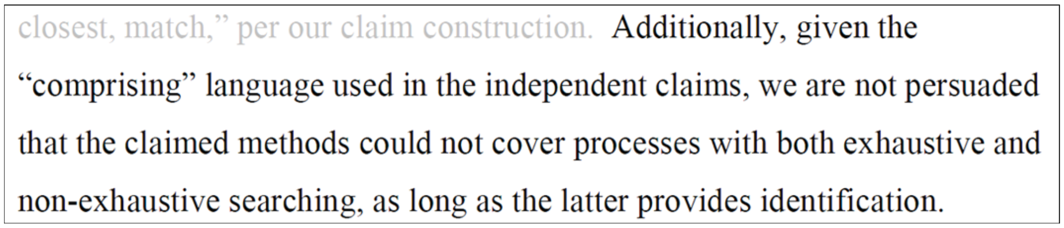
289. The only algorithm Ghias teaches for conducting a search is to compare a query statement against every record in the data set against which the algorithm is to be run—and is thus always an exhaustive search. Accordingly,

---

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:


closest, match," per our claim construction. 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"));

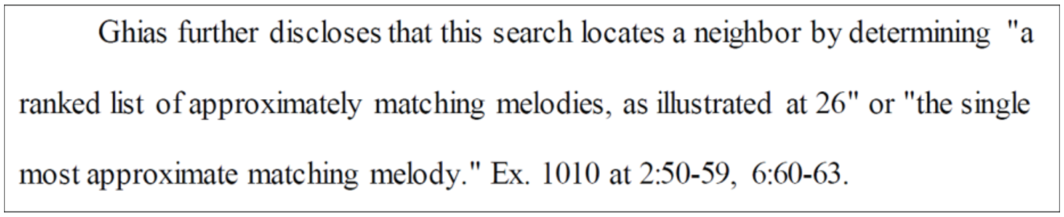
(2) a "ranked list of approximately matching melodies" (Ghias, 2:50-59; Ghias, 6:60-63 ("a list of songs ranked by how well they matched the query"); Moulin Depo. 118:9-22); or

(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. <u>Petition</u>: As support for the claimed "identifying a neighbor," I note that the Petition relies on the following:

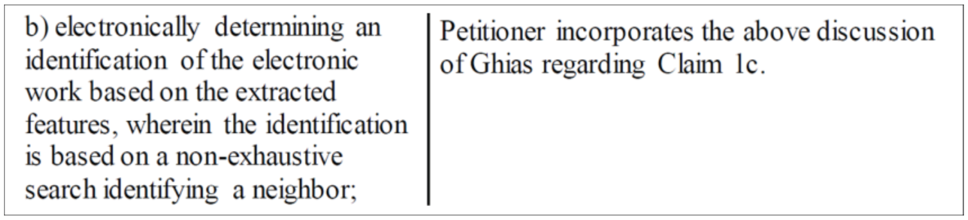


Ghias further discloses that this search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody." Ex. 1010 at 2:50-59, 6:60-63.

Pet. ('988) 10.

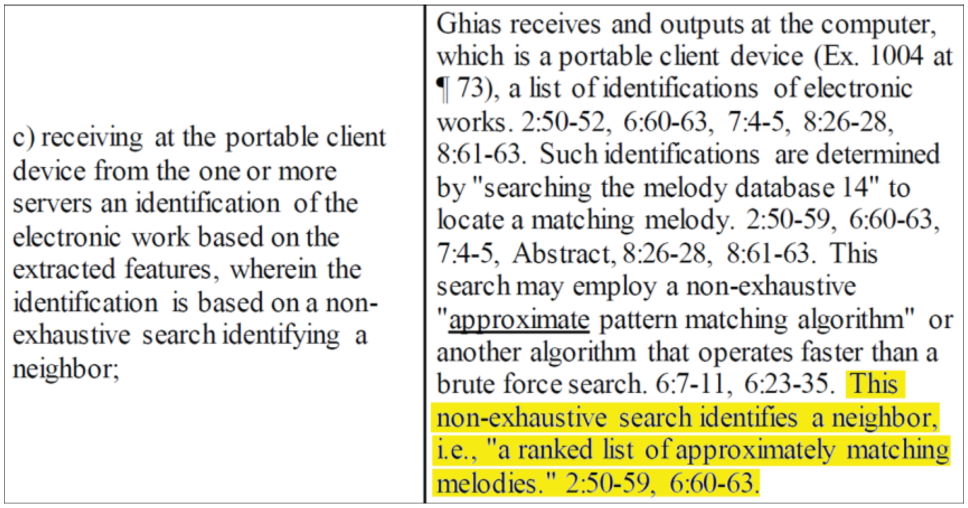
296. <u>Petition Charts</u>: The charts in the Petition the same assertions and passages from Ghias:

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.

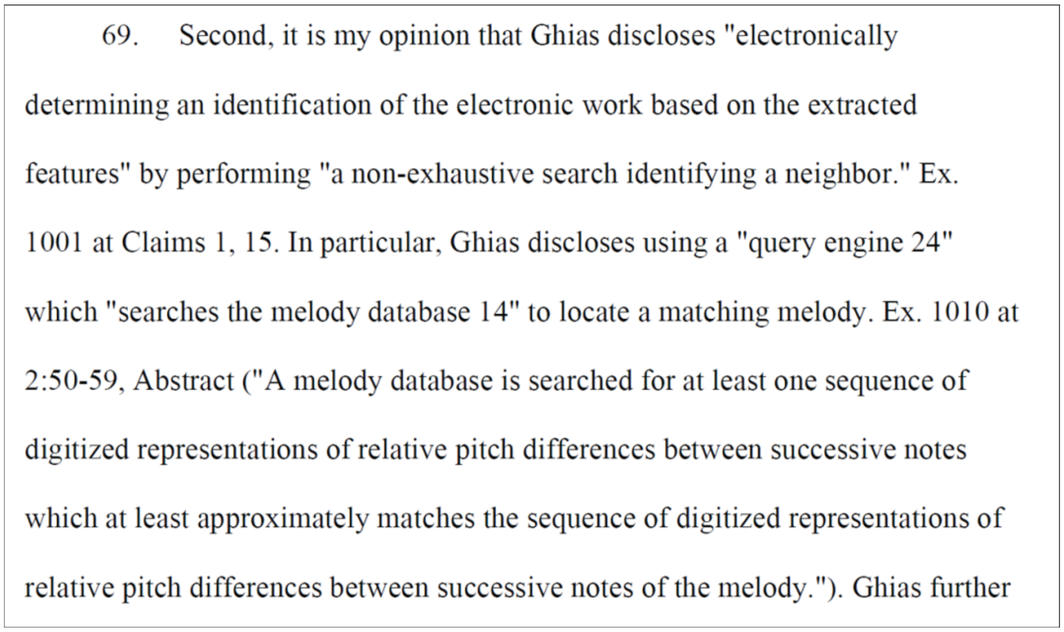
Pet. (‘988) at 14. The chart for claim 1, element [c], in turn, provides:



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.

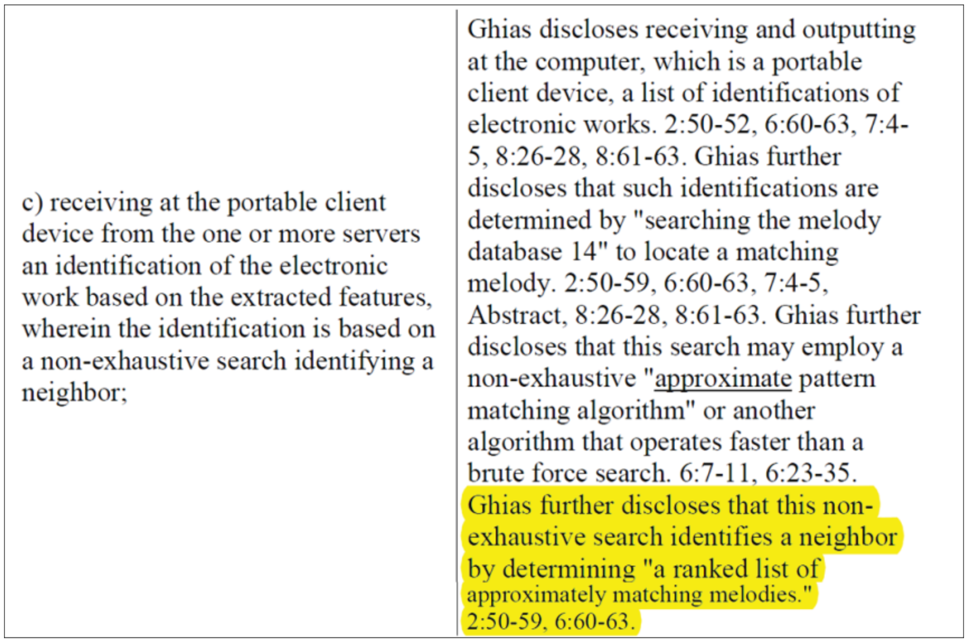
297. Declaration: Petitioner’s Declaration relies on the same assertion and passages from Ghias:



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. <u>Declaration Charts</u>: 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 "<u>approximate</u> 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.

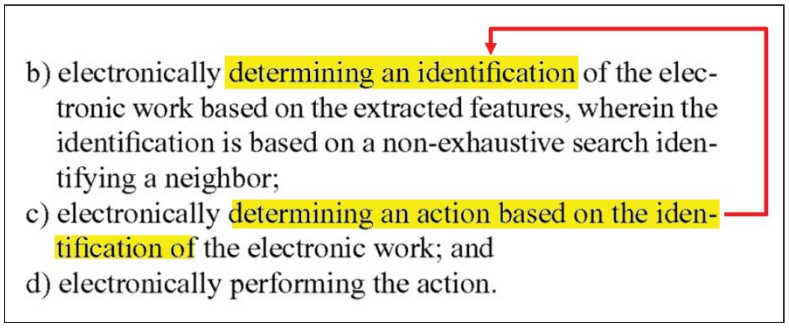
Moulin Decl. (‘988) ¶75.

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;"

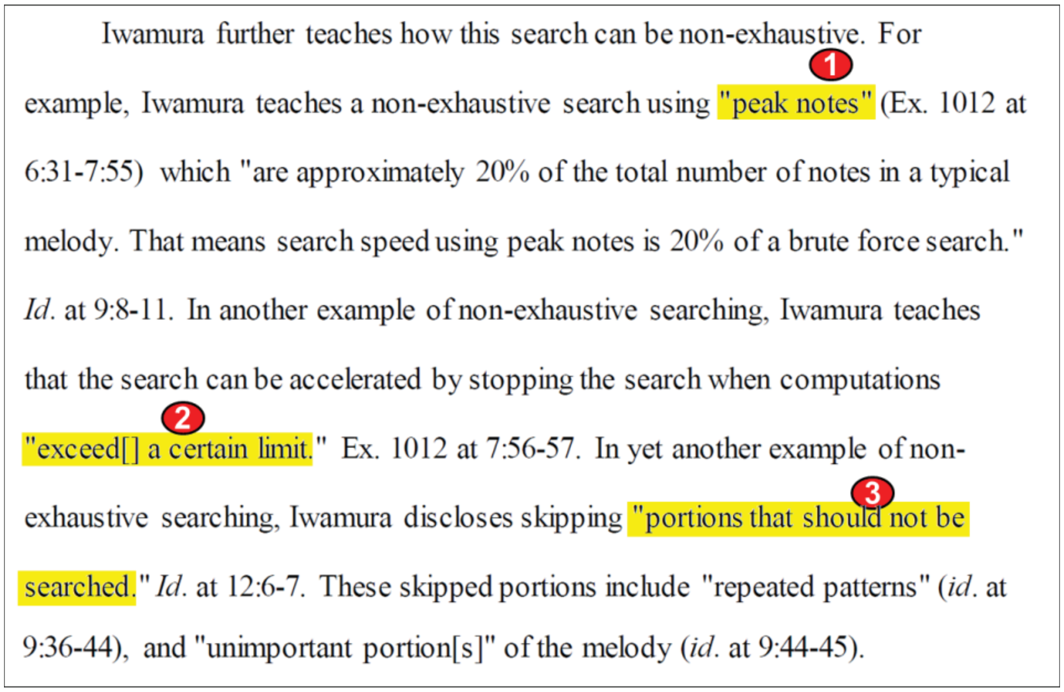
(b) limit function: a search in which a specific comparison of the work to be identified to a specific record in the database "can be stopped," when the specific computation of the total absolute difference between the work to be identified and the specific record exceeds a certain limit;

(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:

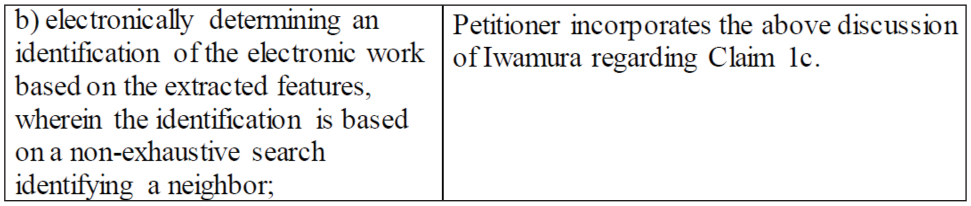
315. Petition:



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).

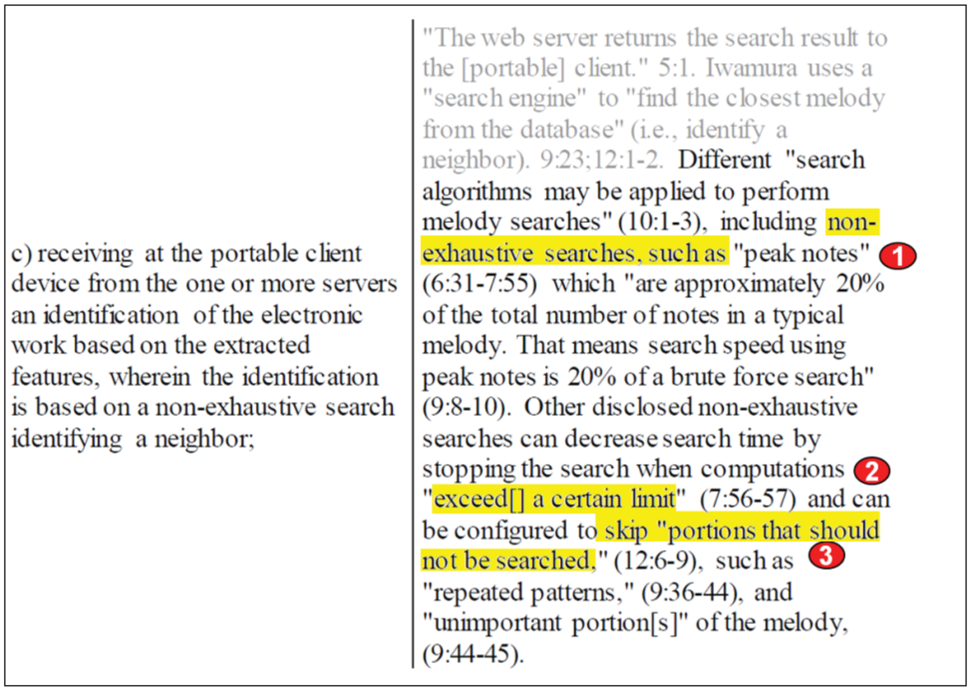
Pet. ('988) at 47-48.

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.

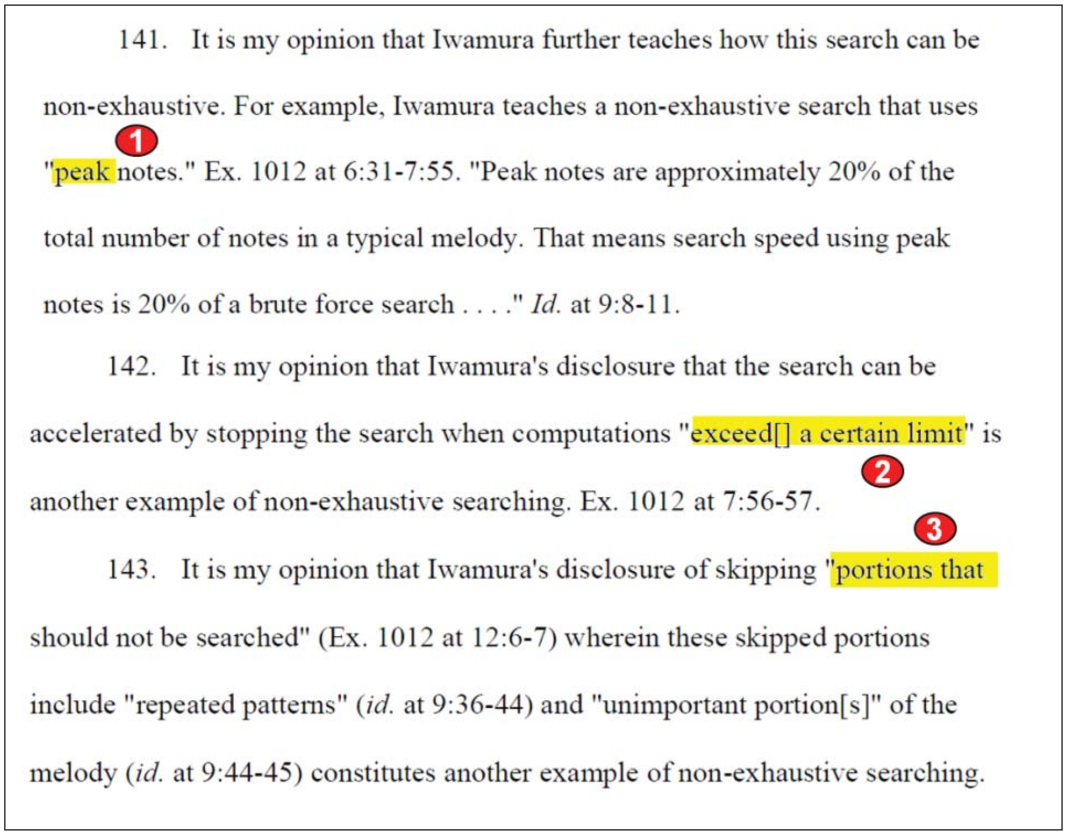
Pet. ('988) at 52. Claim 1, element (c):



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;	"The web server returns the search result to the [portable] client." 5:1. Iwamura uses a "search engine" to "find the closest melody from the database" (i.e., identify a neighbor). 9:23;12:1-2. Different "search algorithms may be applied to perform melody searches" (10:1-3), including non-exhaustive searches, such as "peak notes" ① (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 ② "exceed[] a certain limit" (7:56-57) and can be configured to skip "portions that should not be searched," (12:6-9), such as ③ "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.

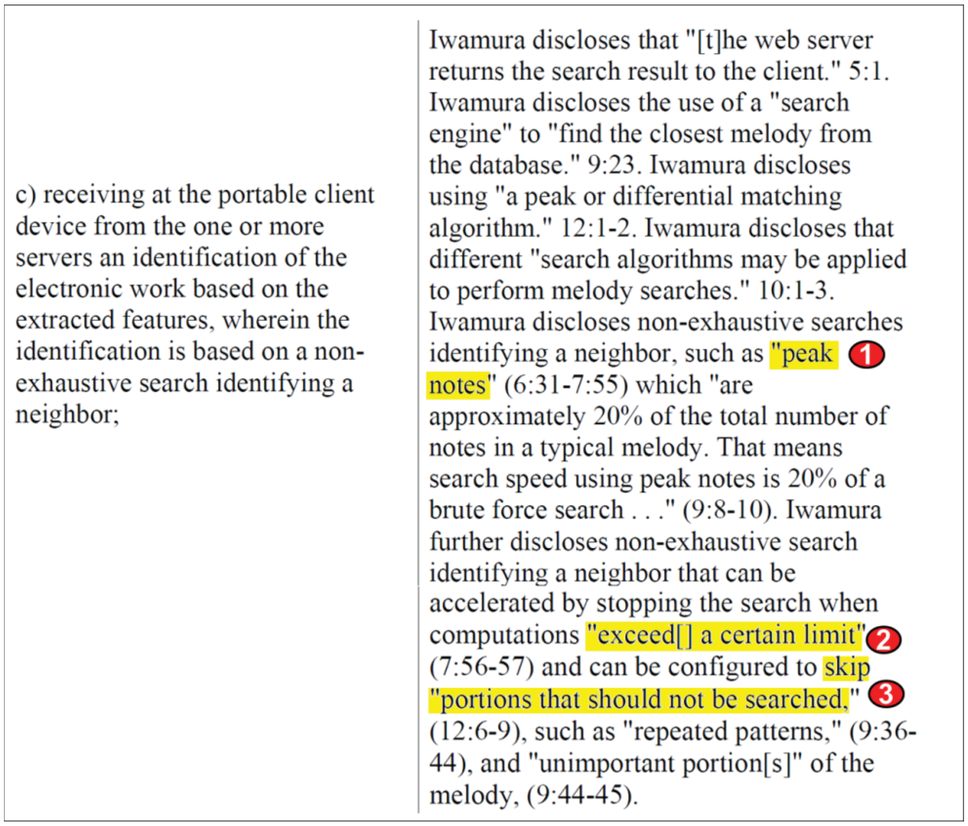
Moulin Decl. ('988) ¶¶141-143.

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):



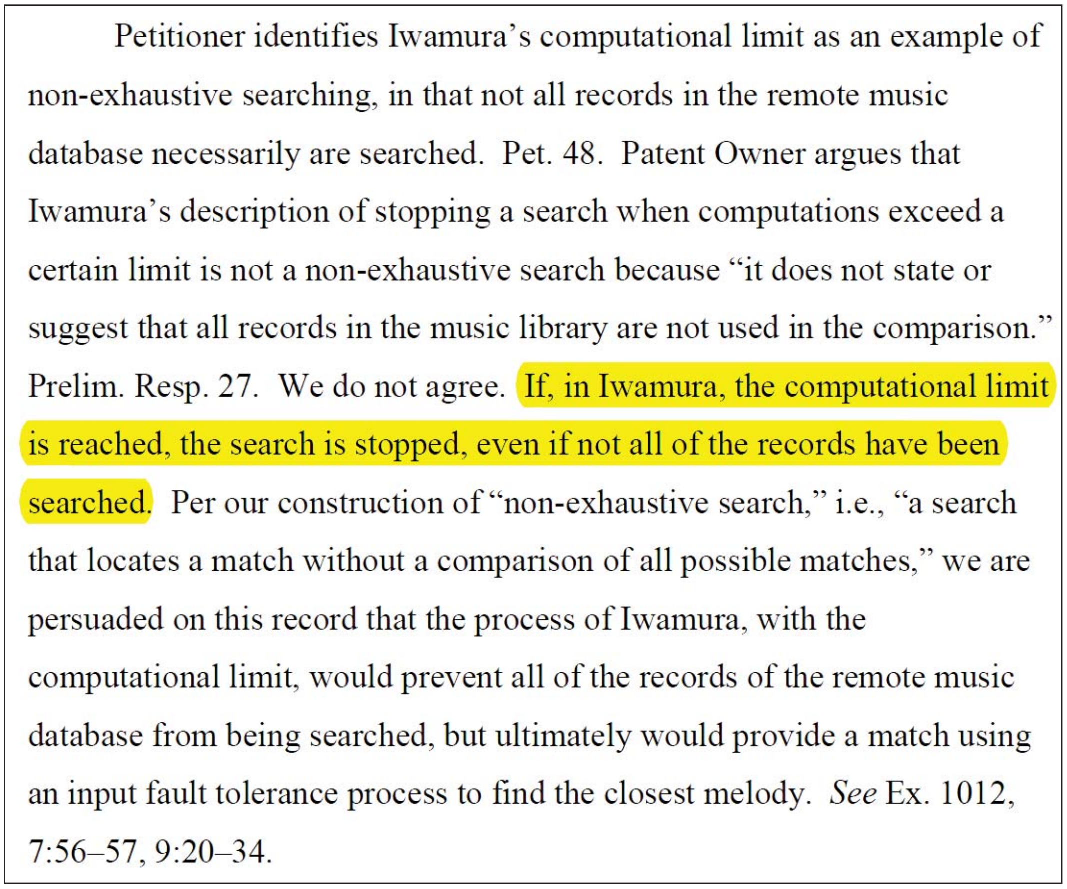
c) receiving at the portable client device from the one or more servers an identification of the electronic work based on the extracted features, wherein the identification is based on a non-exhaustive search identifying a neighbor;	Iwamura discloses that "[t]he web server returns the search result to 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 ① notes" (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"② (7:56-57) and can be configured to skip "portions that should not be searched," ③ (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. <u>Board’s concerns</u>: 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

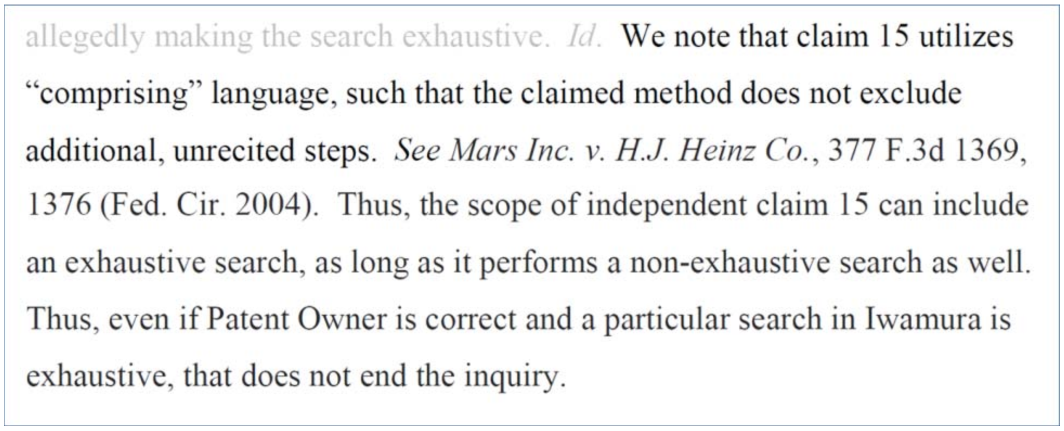
(b) stopping the search process altogether.

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:


allegedly making the search exhaustive. *Id.* 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.

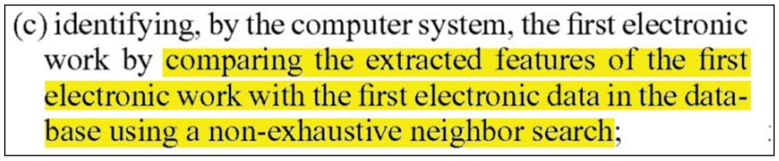
328. The Board instituted the '179 IPR on two grounds:

- 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.

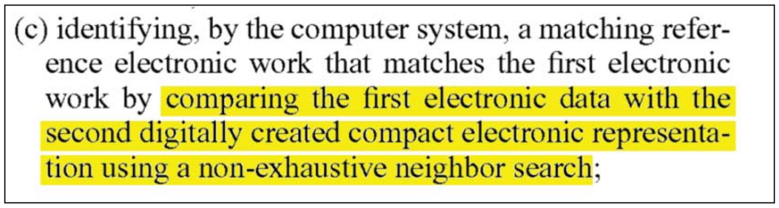
Decision ('179) at 15. I address each Ground in turn.

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

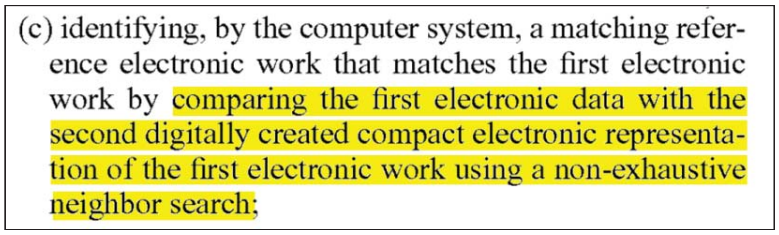
329. I understand that to anticipate a claim, all elements of the claim need to be disclosed in a single prior art reference—in this case, Conwell. Each independent claim of the '179 patent includes a limitation "comparing [the extracted features of the work to be identified with extracted features of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

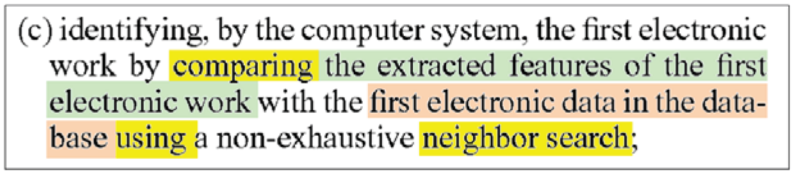
Claim 25: "comparing"

[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:

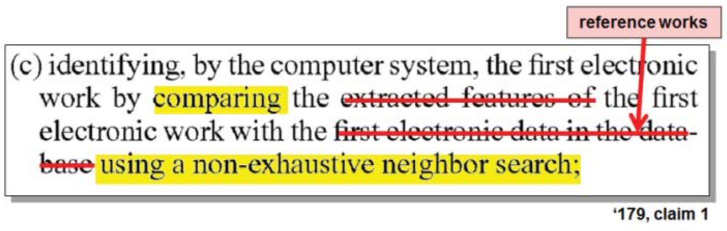
[1] "one or more reference electronic works" (*i.e.*, the references in the database); with

[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:


(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 data-base~~ using a non-exhaustive neighbor search;

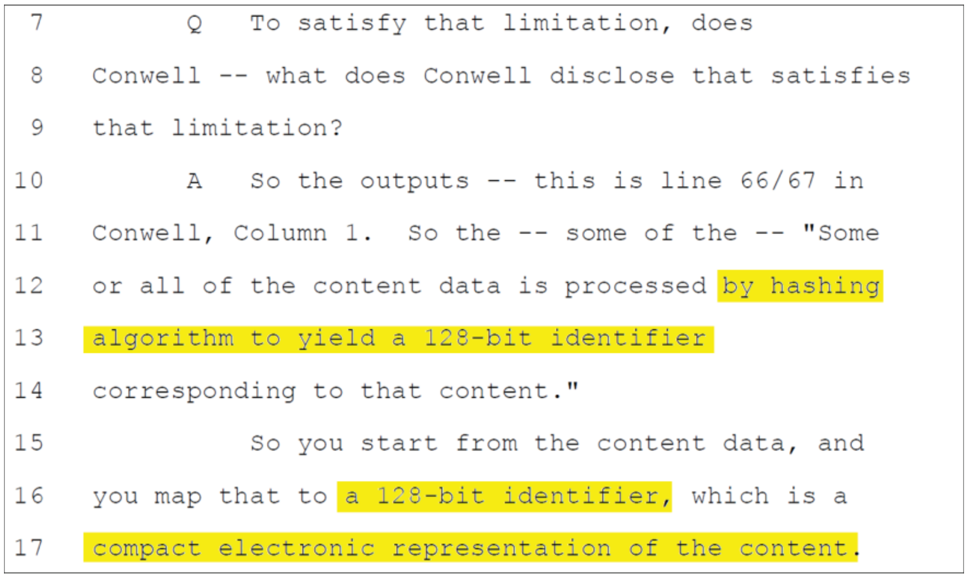
reference works
'179, claim 1

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

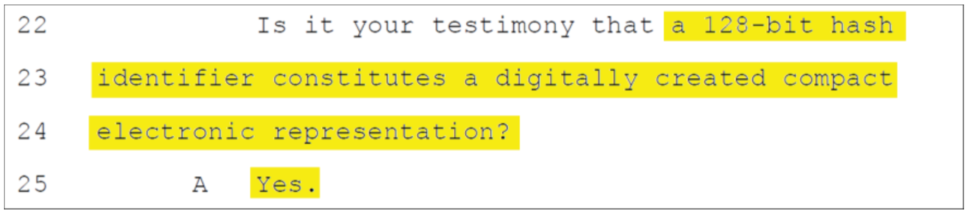
search. *See* Pet. (‘179) at 23 (the hashed “identifier extracted from a reference electronic work” are compared to the hashed “extract identifiers from content”); Moulin Decl. (‘179) ¶85 (“Conwell further teaches that a hash algorithm can be selected so that ‘similar, but non-identical, inputs map on 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.

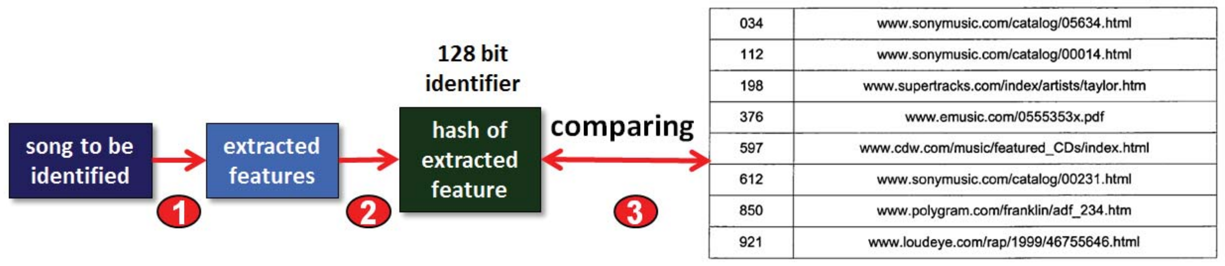
Moulin Depo. 180:7-17.


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:



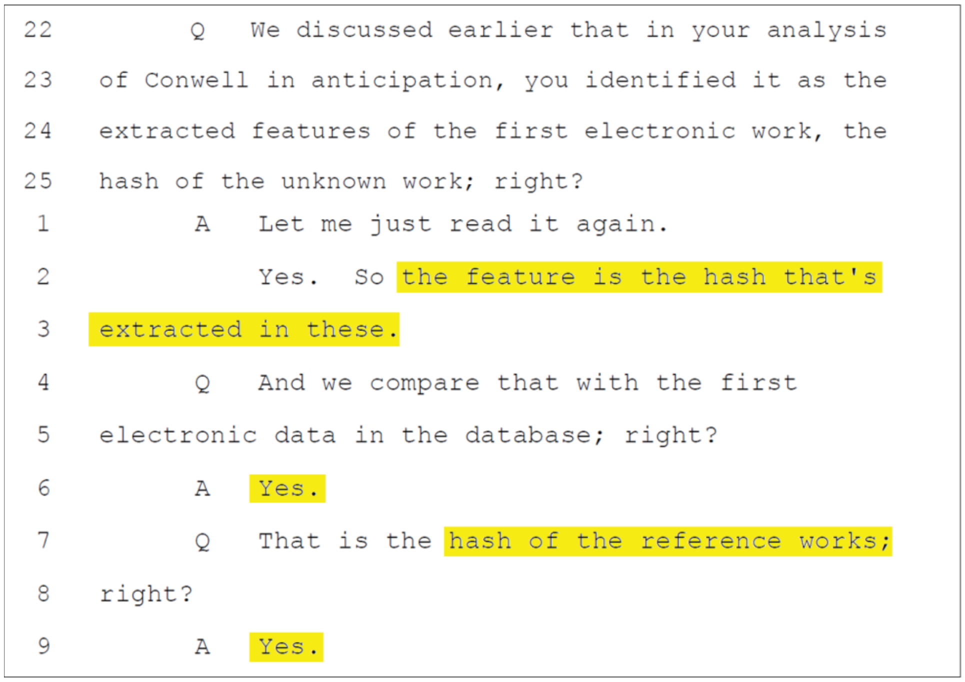
034	www.sonymusic.com/catalog/05634.html
112	www.sonymusic.com/catalog/00014.html
198	www.supertracks.com/index/artists/taylor.htm
376	www.emusic.com/0555353x.pdf
597	www.cdw.com/music/featured_CDs/index.html
612	www.sonymusic.com/catalog/00231.html
850	www.polygram.com/franklin/adf_234.htm
921	www.loudeye.com/rap/1999/46755646.html

FIG. 3

[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.

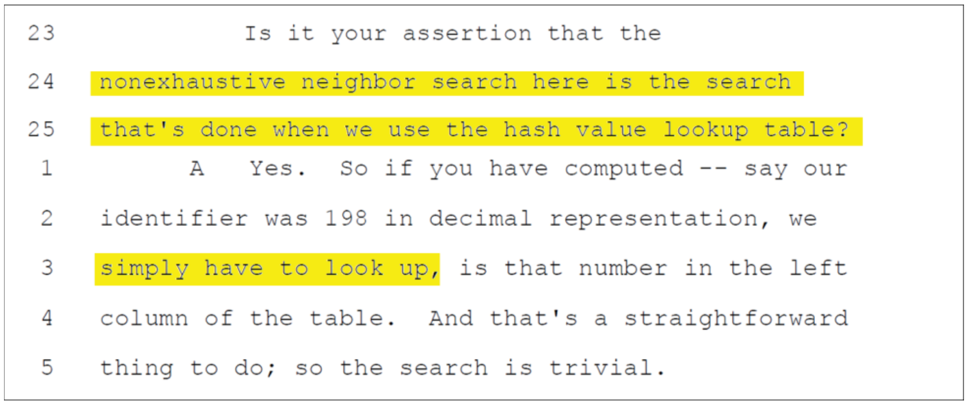
Q And we compare that with the first electronic data in the database; right?

A Yes.

Q That is the hash of the reference works; right?

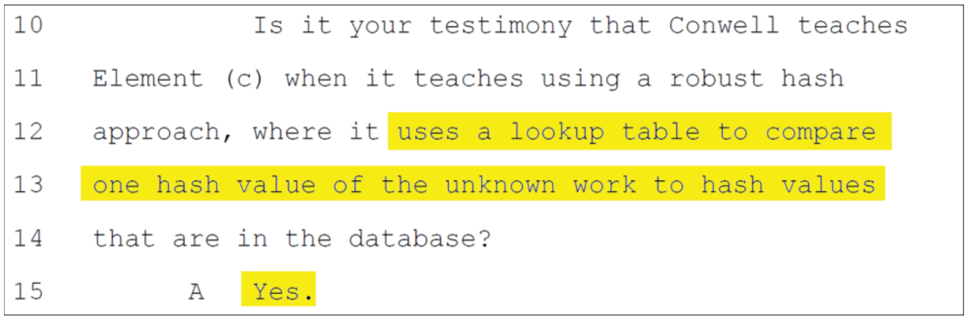
A Yes.

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.

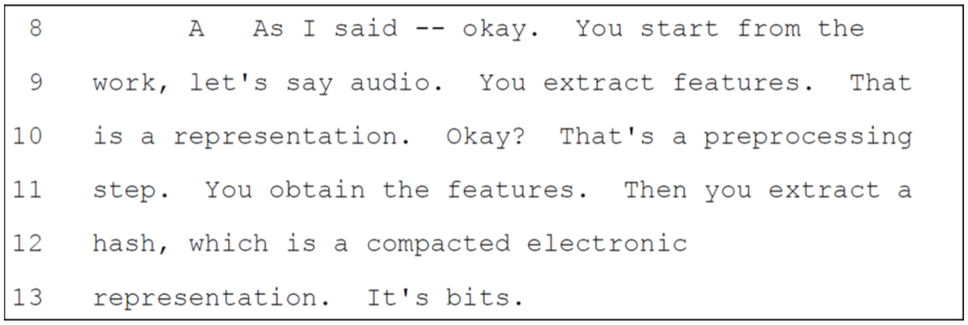
Moulin Depo. 199:23-200:5; 191:8-12; 173:25-174:6; 170:14-18; 185:20-24.


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):



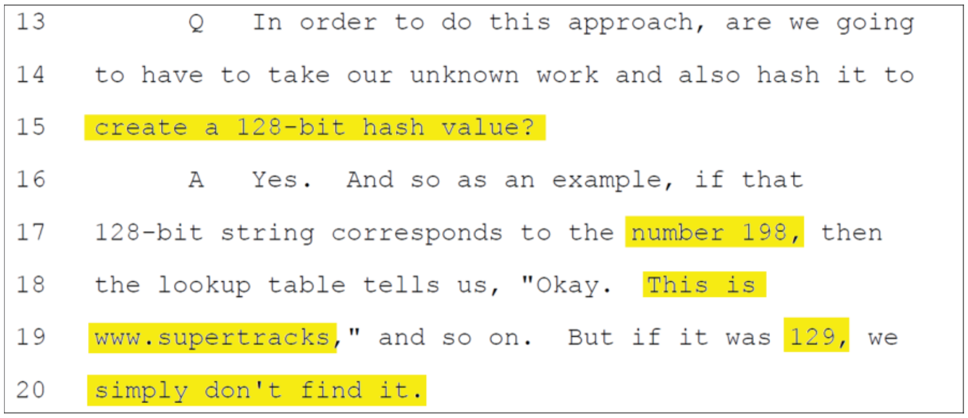
A As I said -- okay. You start from the work, let's say audio. You extract features. That is a representation. Okay? That's a preprocessing step. You obtain the features. Then you extract a hash, which is a compacted electronic representation. It's bits.

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.

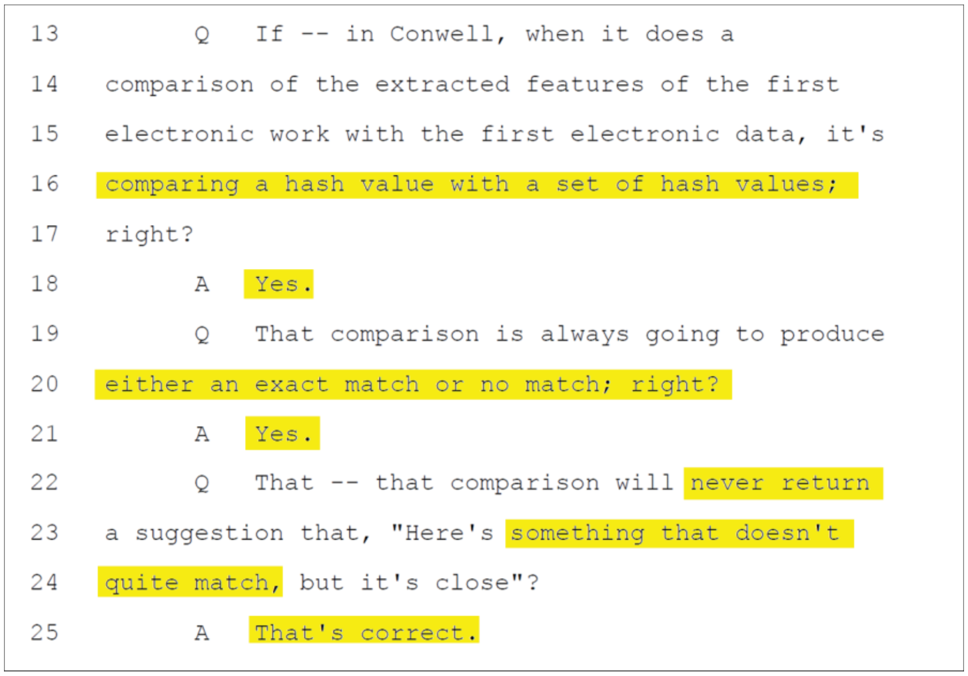
342. The "lookup table" disclosed in Conwell looks for an exact match of the identifiers. If there is an exact match, the search identifies that match. If there is a "neighbor"—*i.e.*, "a close, but not necessarily exact or closest, match," the search will not identify such a neighbor. A neighbor search—a 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 198, then
the lookup table tells us, "Okay.  This is
www.supertracks," and so on.  But if it was 129, we
simply don't find it.
```

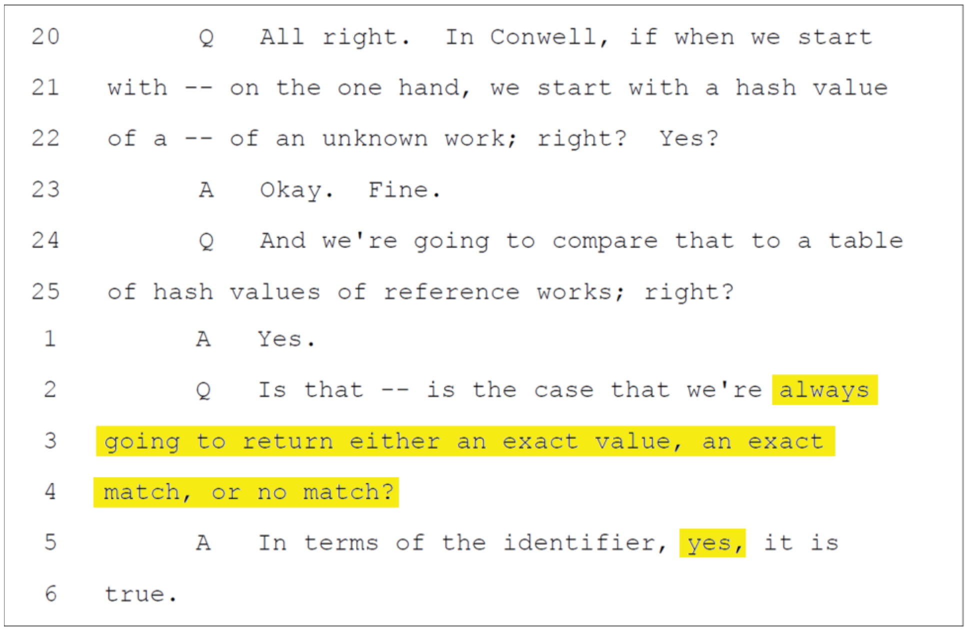
Moulin Depo. 188:13-20.

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.

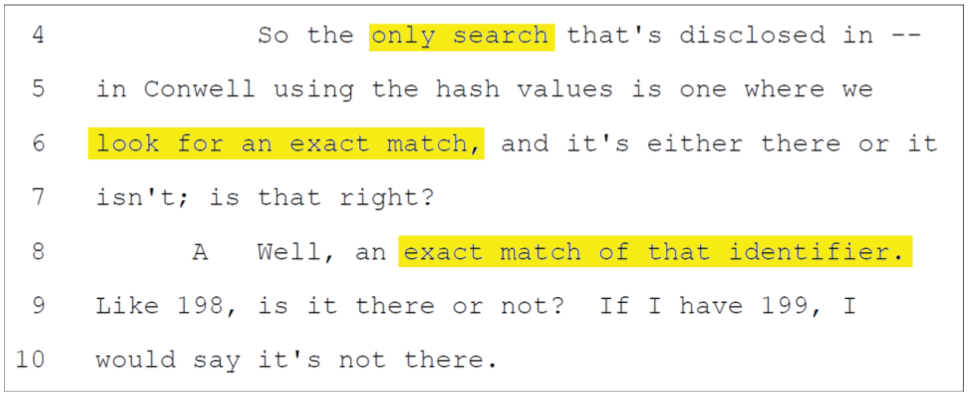
Q And we're going to compare that to a table of hash values of reference works; right?

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:

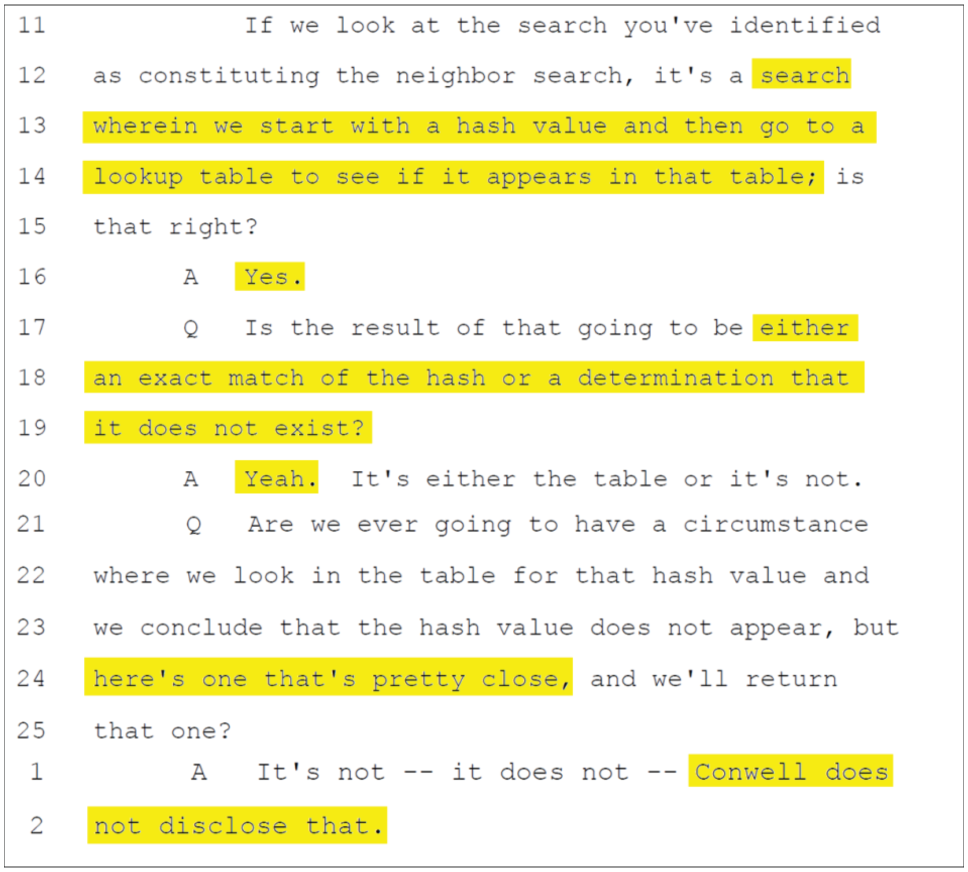

So the only search that's disclosed in -- in Conwell using the hash values is one where we look for an exact match, and it's either there or it isn't; is that right?

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.



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.

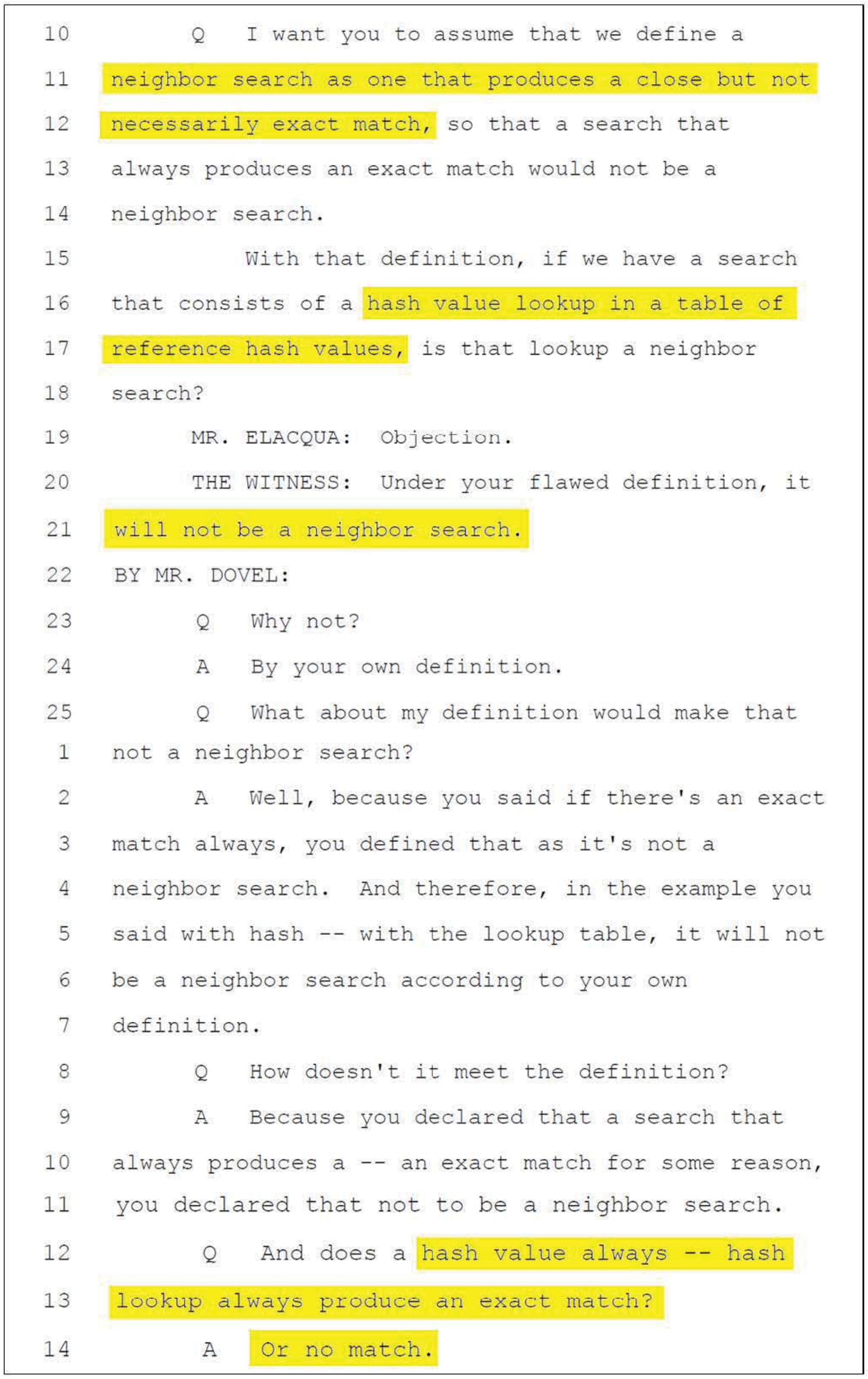
Q Is the result of that going to be either an exact match of the hash or a determination that it does not exist?

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?

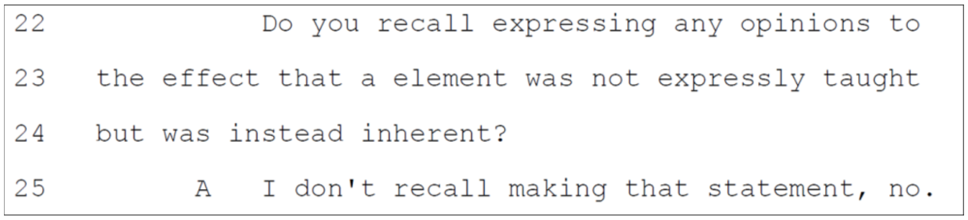
A Because you declared that a search that always produces a -- an exact match for some reason, you declared that not to be a neighbor search.

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:

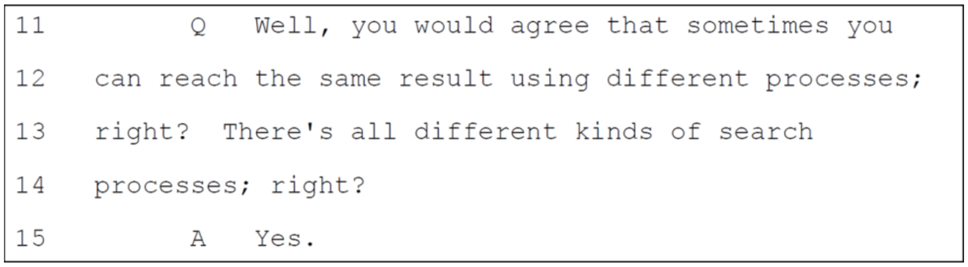


Do you recall expressing any opinions to the effect that a element was not expressly taught but was instead inherent?

A I don't recall making that statement, no.

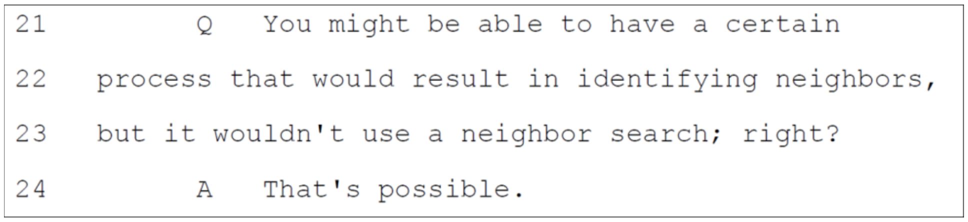
346. While the search disclosed in Conwell may have the result of identifying a neighbor of the work to be identified, it does not do so by "comparing [the extracted features] using [a] neighbor search" as required by the '179 claims. It is my understanding that if a process disclosed in a reference 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.

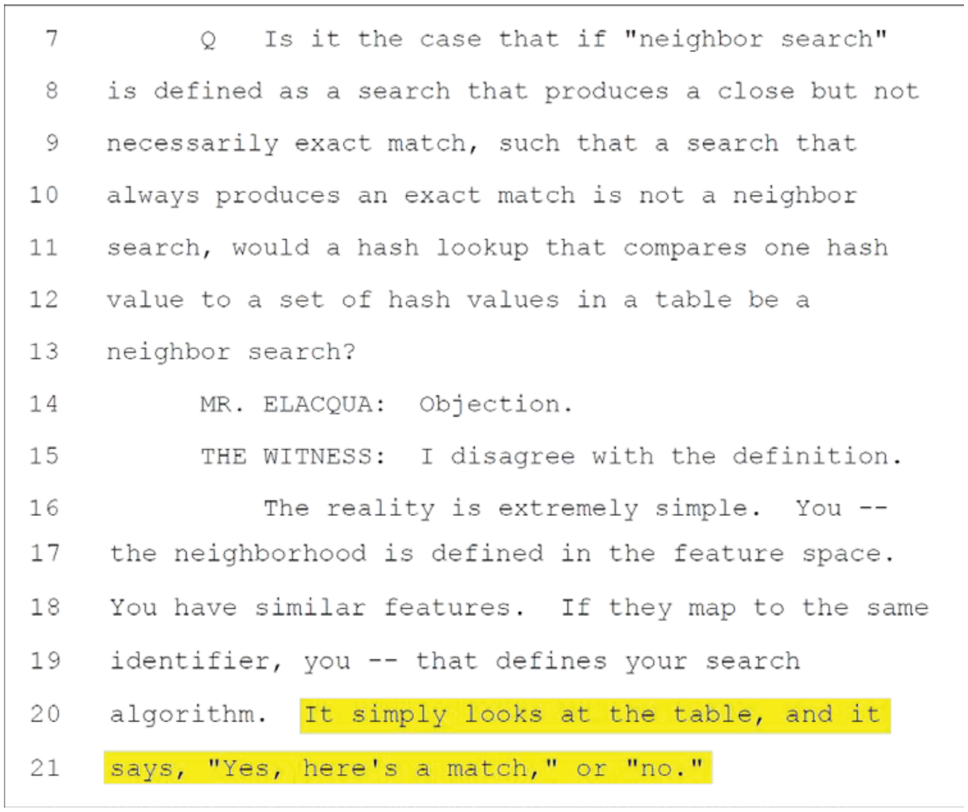
Moulin Depo. 266:11-15.


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, such that a search that always produces an exact match is not a neighbor search, would a hash lookup that compares one hash value to a set of hash values in a table be a neighbor search?

MR. ELACQUA: 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."

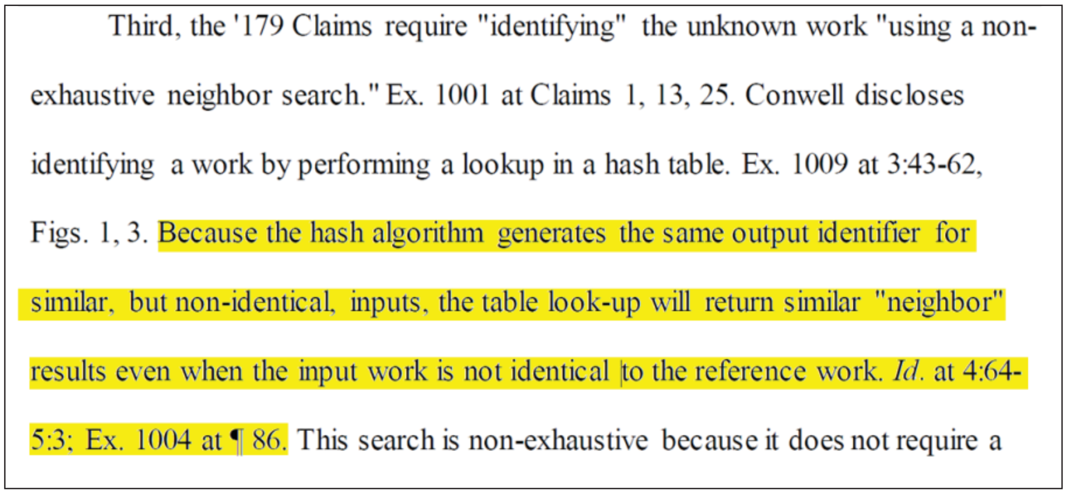
Moulin Depo. 260:7-21.

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.”

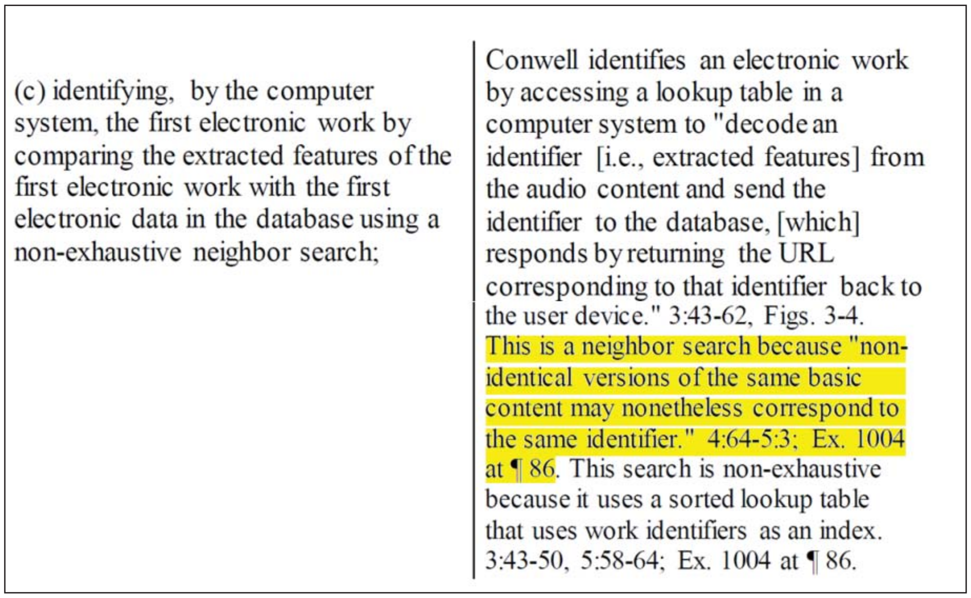
350. Petition: As support that Conwell discloses the claimed “comparing [the extracted features] using [a] neighbor search,” the Petition relies on the

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.

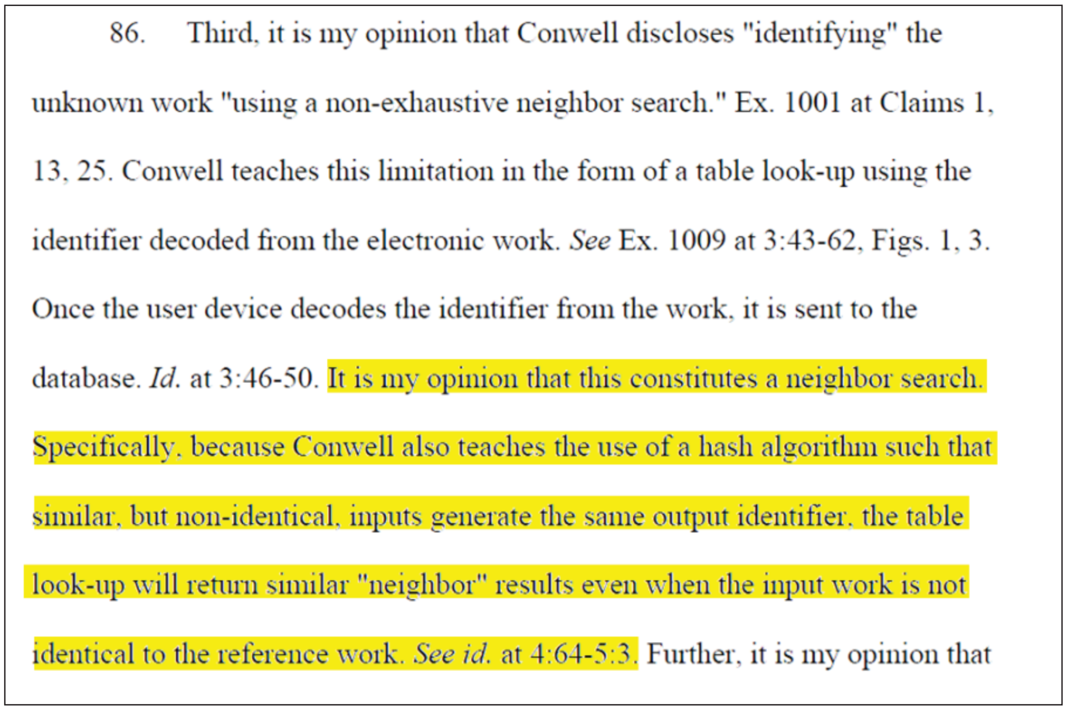
351. Petition chart: The chart in the Petition makes the same assertion and cites the same passage:



(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.

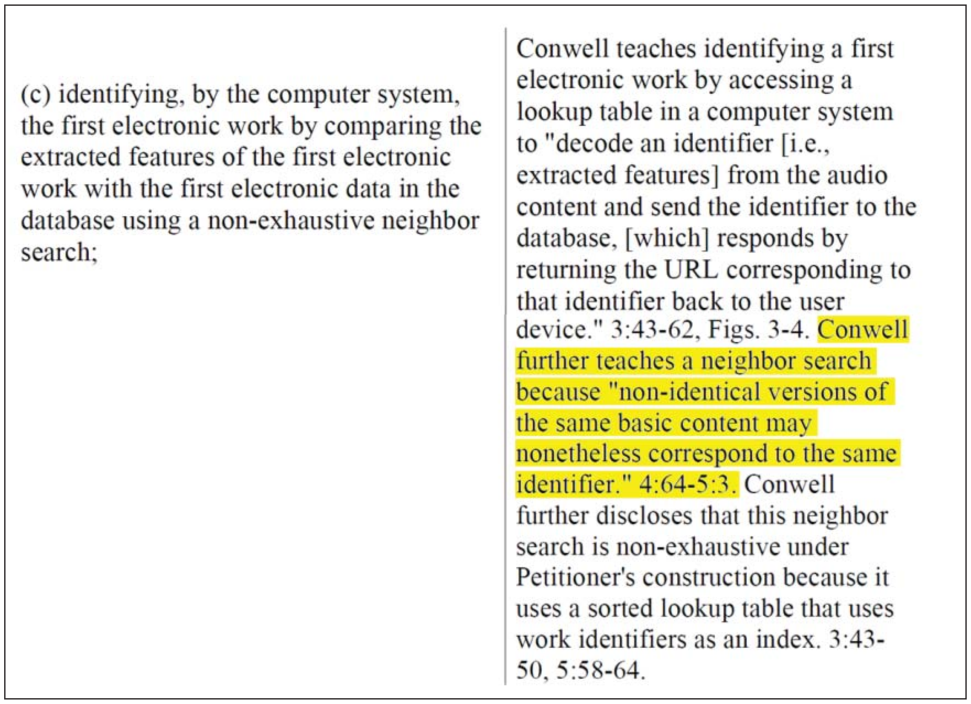
352. Declaration: Petitioner's Declarant makes the same assertion and cites the same passage:



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

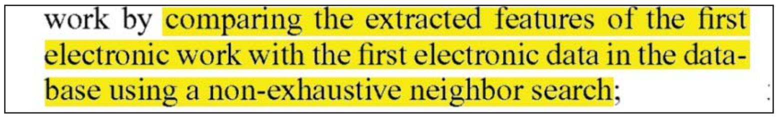
353. Declarant chart: Finally, the chart in Petitioner's Declaration makes the same assertion and cites the same passage:



(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.

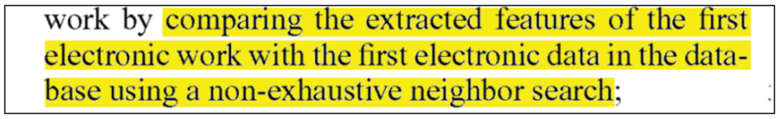
354. The Petition, Declaration, and corresponding charts fail to establish that Conwell teaches the claimed “comparing [the extracted features] using [a] neighbor search.” In fact, the Petition, Declaration, and charts fail to address the actual claim language. As set forth above, the claims require



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 database 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:

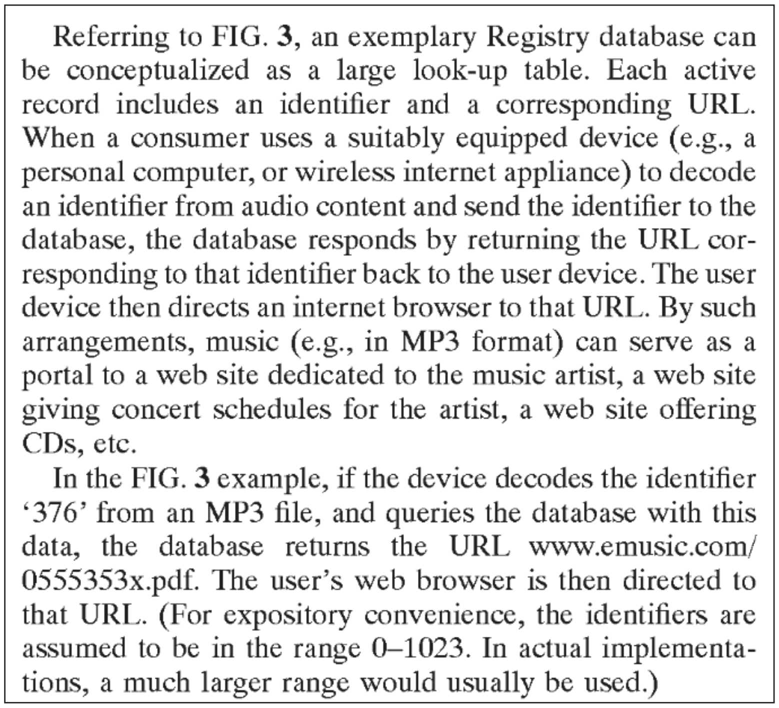
> "Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar 'neighbor' results even when the input work is not identical to the reference work."

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. <u>Conwell, 3:43-62</u>:

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

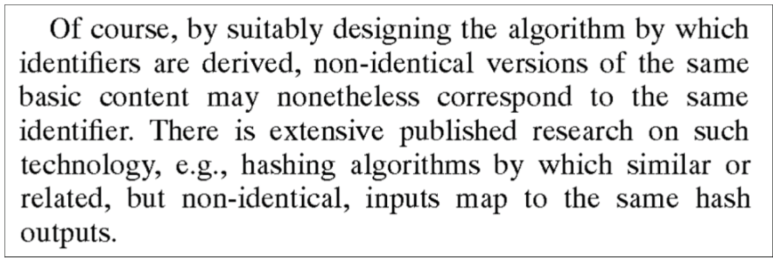


Referring to FIG. **3**, an exemplary Registry database can be conceptualized as a large look-up table. Each active record includes an identifier and a corresponding URL. When a consumer uses a suitably equipped device (e.g., a personal computer, or wireless internet appliance) to decode an identifier 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."

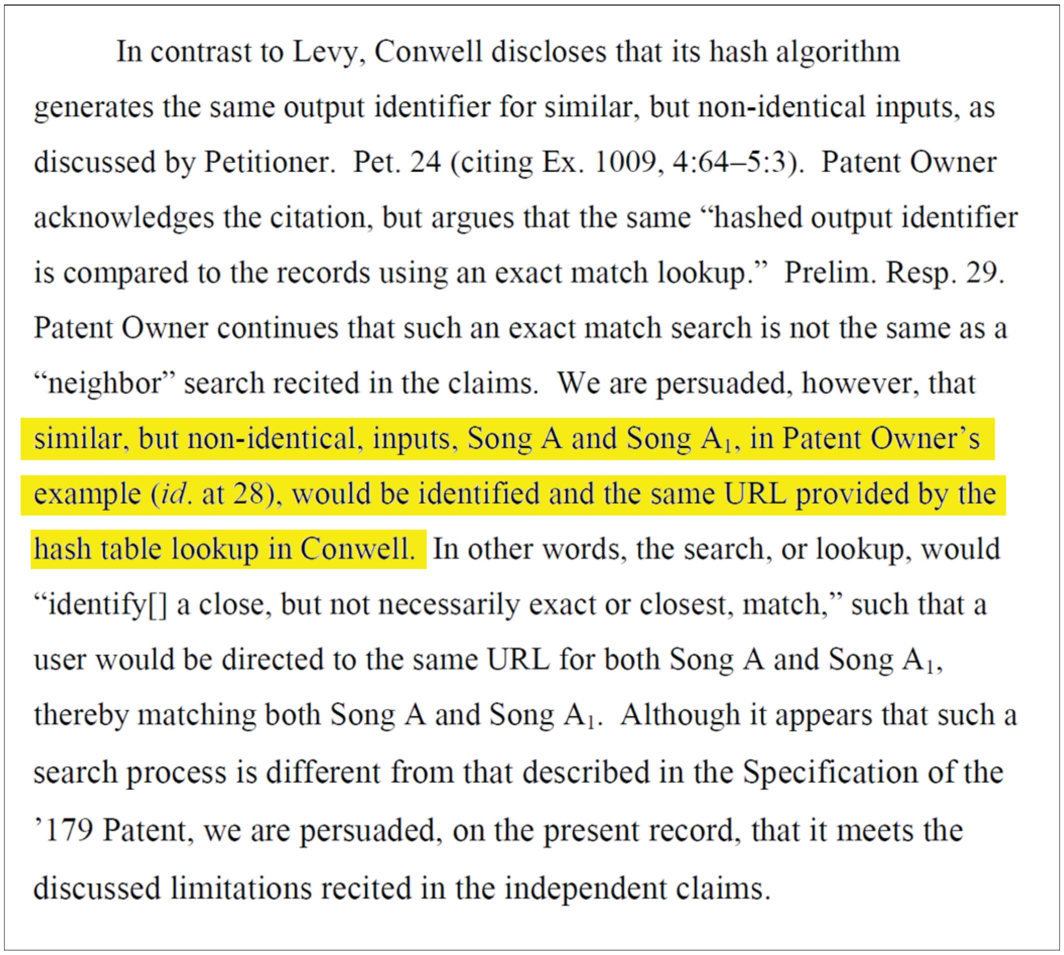
359. Conwell, 4:64-5:3:



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.

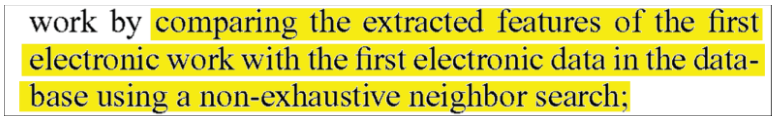
One skilled in the art would understand that this cited passage states that similar works might correspond to the same hash. Using Figure 3 to illustrate, this passage suggests that two similar works might produce the same hash, e.g., the 4th entry in the table "376." The passage does not state that 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 data-base 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.

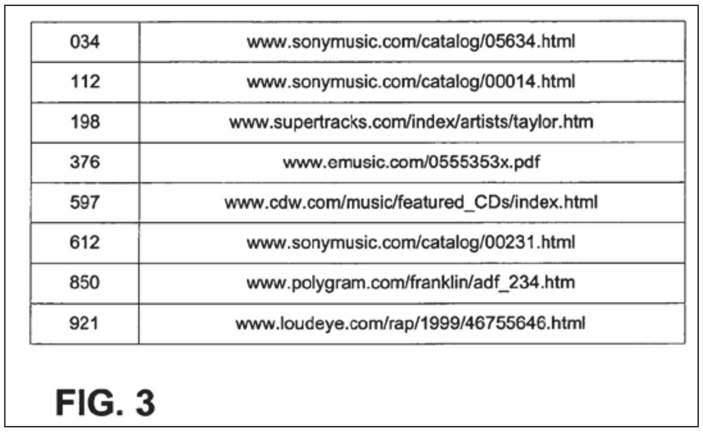
364. As I explained above, the claimed “non-exhaustive” search is “a search that locates a match without a comparison of all possible matches.” Section V(B); Decision (‘179) at 7. A non-exhaustive search uses an intelligent algorithm to reduce the number of potential matches. By contrast, an exhaustive 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. <u>Express</u>: 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.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

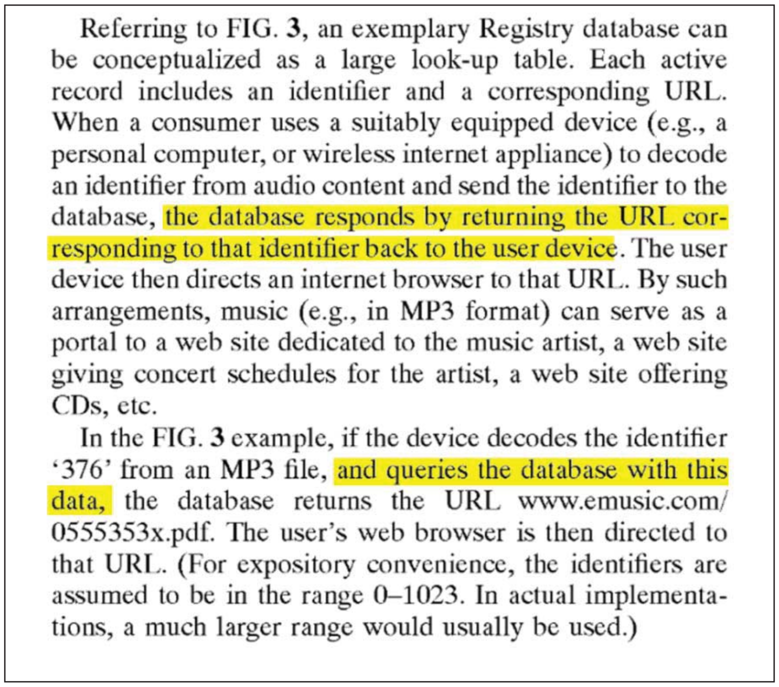
FIG. 3

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.

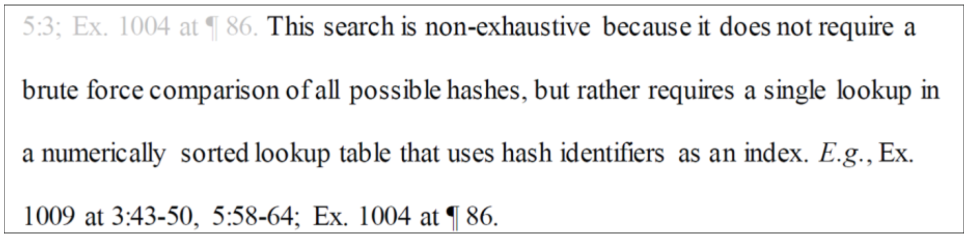
370. Inherent: Conwell also does not inherently disclose using the "look-up table" to conduct a non-exhaustive search. First, as a preliminary matter, I note that neither the Petition nor the Petitioner's Declarant relied on any theory that Conwell inherently discloses a non-exhaustive search. Pet. ('179) 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.

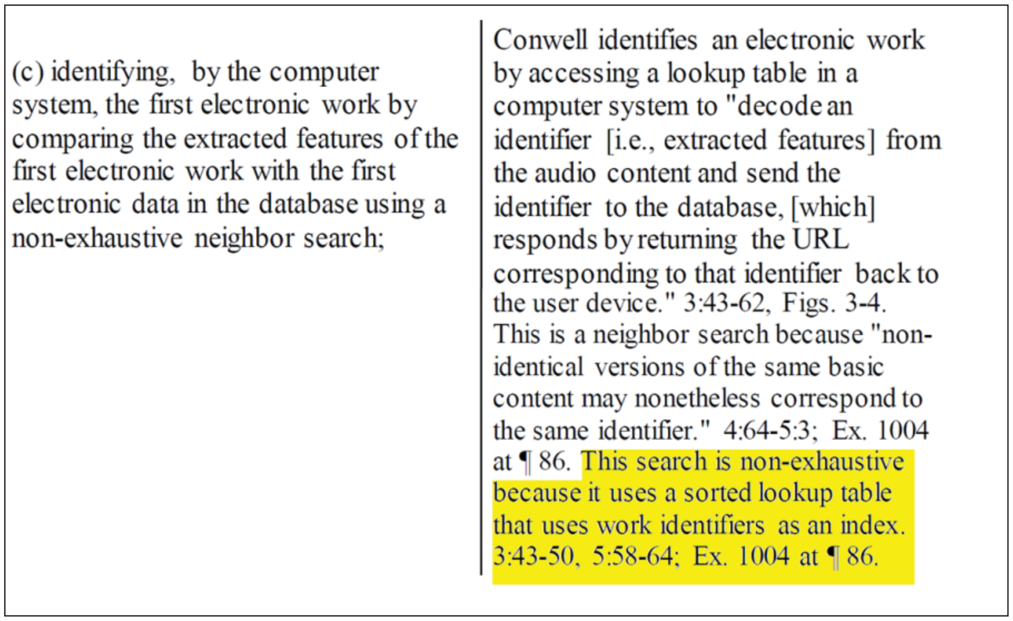
372. The Petition, Declaration, and corresponding charts fail to demonstrate that Conwell teaches "comparing [the extracted features] using a non-exhaustive… search."

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


5:3; Ex. 1004 at ¶ 86. 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.

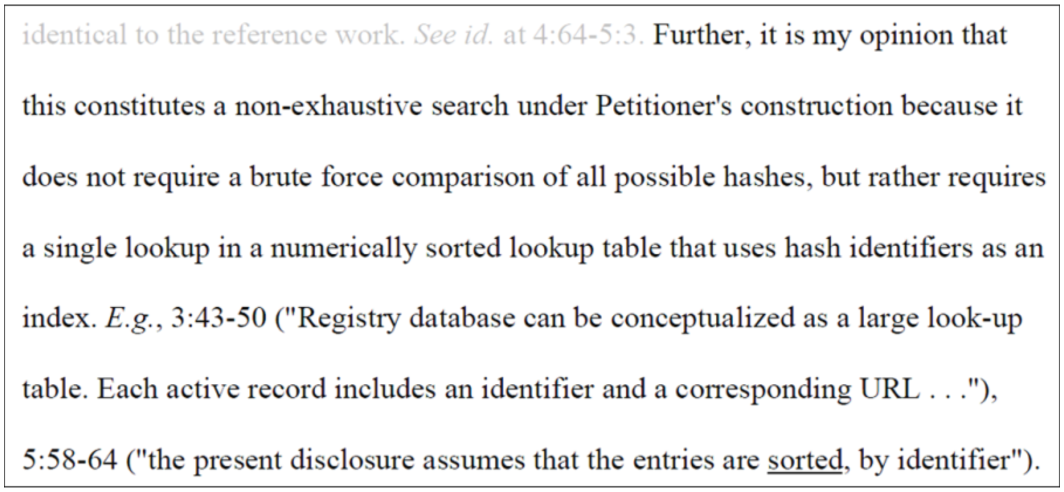
374. Petition chart: The chart in the Petition makes the same assertion and cites the same two passages from Conwell:



(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.

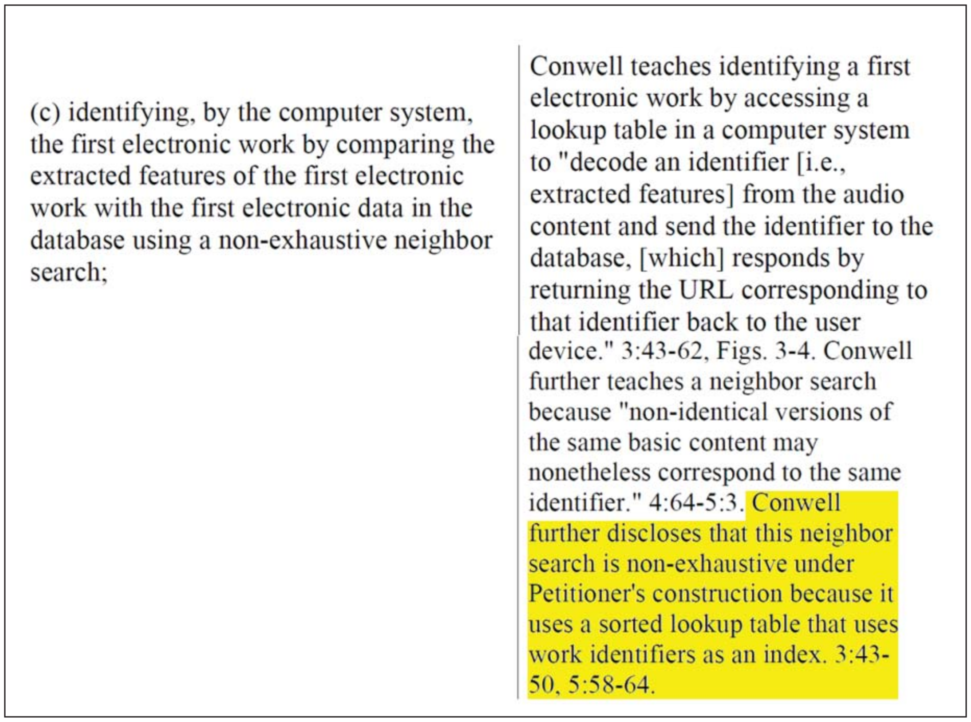
375. Declaration: Petitioner’s Declarant makes the same assertion and relies on the same two passages from Conwell:



identical to the reference work. *See id.* at 4:64-5:3. 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:



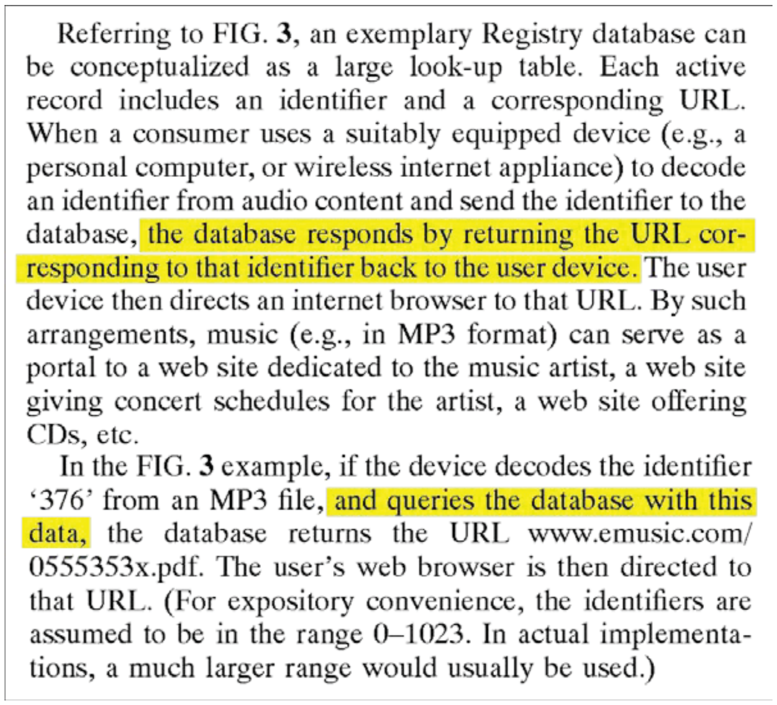
(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Conwell teaches identifying a first electronic work by accessing a lookup table in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

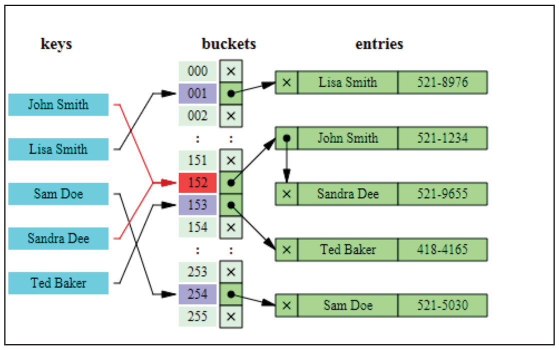
In the FIG. **3** example, if the device decodes the identifier '376' from an MP3 file, and queries the database with this data, the database returns the URL www.emusic.com/0555353x.pdf. The user's web browser is then directed to that URL. (For expository convenience, the identifiers are assumed to be 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….



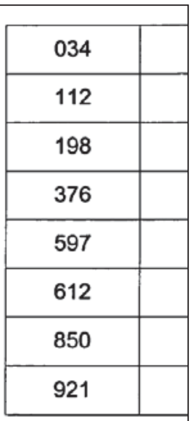
https://en.wikipedia.org/wiki/Hash_table.[37] The hash table can be looked up directly because there is a line item for every possible number. For example, if the query hashes to a value of 5, the search could go directly to the 5th record.[38]

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.



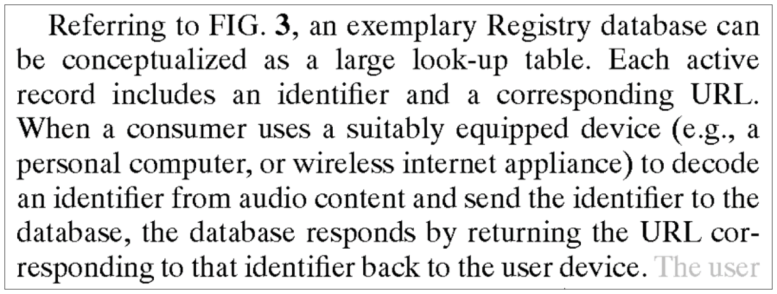
034	
112	
198	
376	
597	
612	
850	
921	

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.

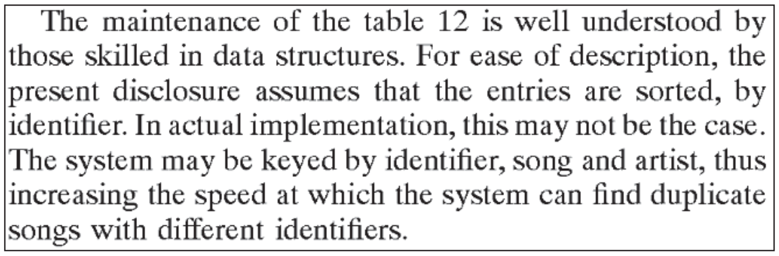
382. The two cited passages from Conwell relied on in the Petition, Declaration, and charts do not disclose any particular search algorithm using the sorted lookup-table, much less an algorithm that is a non-exhaustive search. Each passage from Conwell is addressed in turn.

383: <u>Passage 1</u>:


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

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. Passage 2:


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.

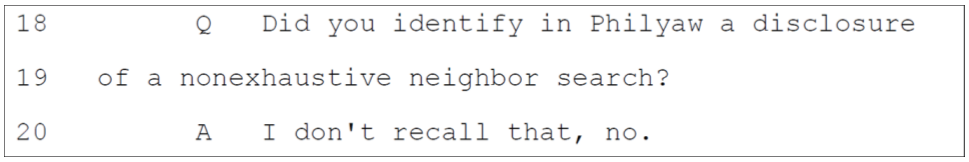
Conwell, 5:58-64. Just like the first passage, this passage also does not disclose the claimed "non-exhaustive search." Rather, this passage states that the (1) the maintenance of the table 12 is well understood by those skilled in data structures, (2) the entries can be sorted by identifier, and (3)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

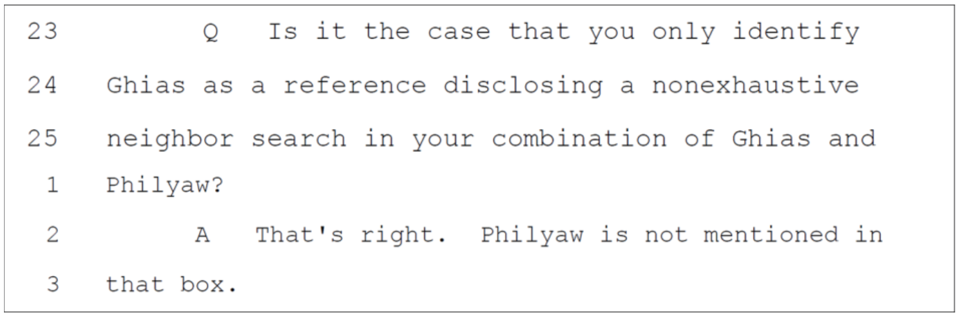
B. '179 Ground 2: The instituted claims of the '179 Patent are not obvious in view of Ghias and Philyaw.

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.

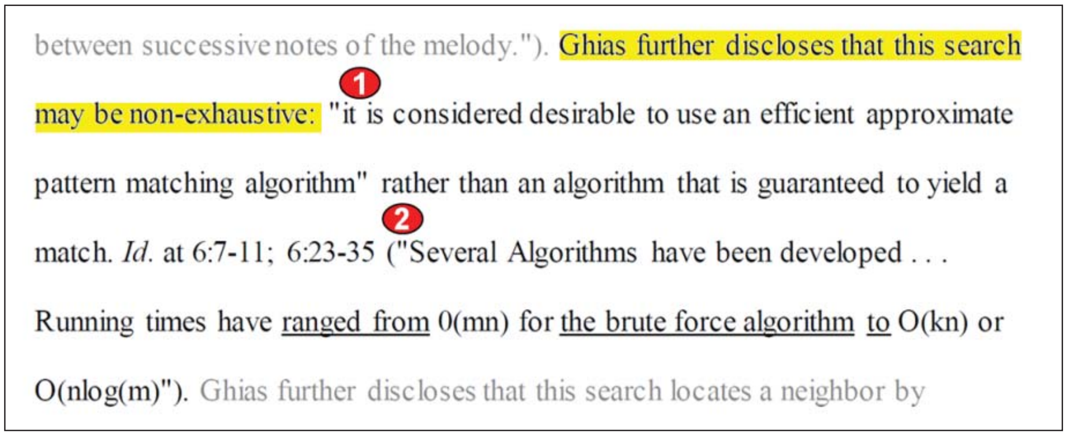
1. non-exhaustive search (claims 1, 13, 25).

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.

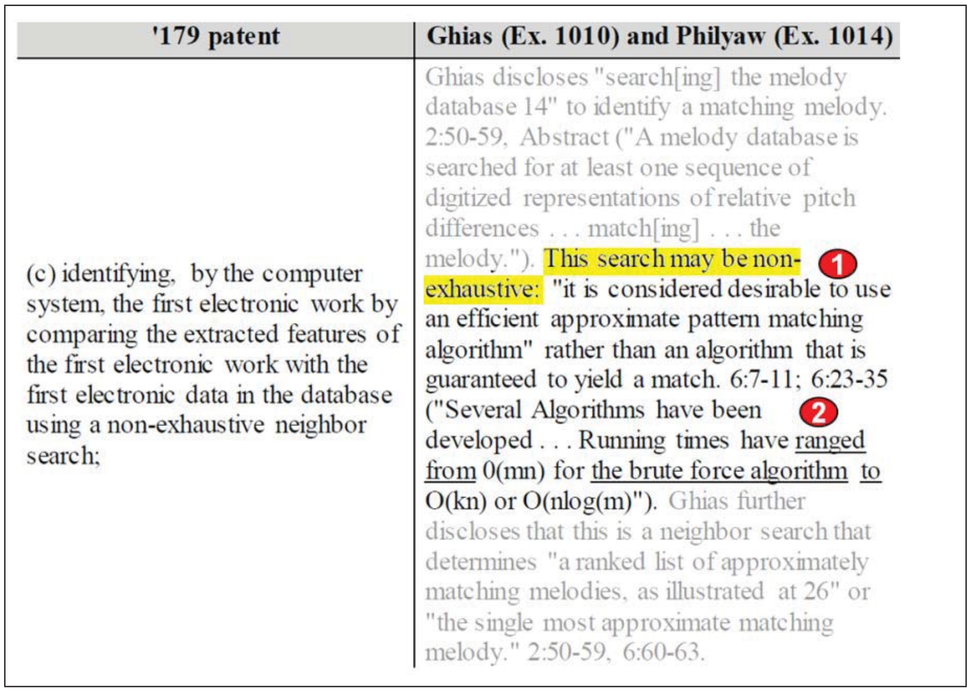
389. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a non-exhaustive search.

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


between successive notes of the melody."). 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 ranged from 0(mn) for the brute force algorithm to O(kn) or O(nlog(m)"). Ghias further discloses that this search locates a neighbor by

Pet. ('179) 47-48.

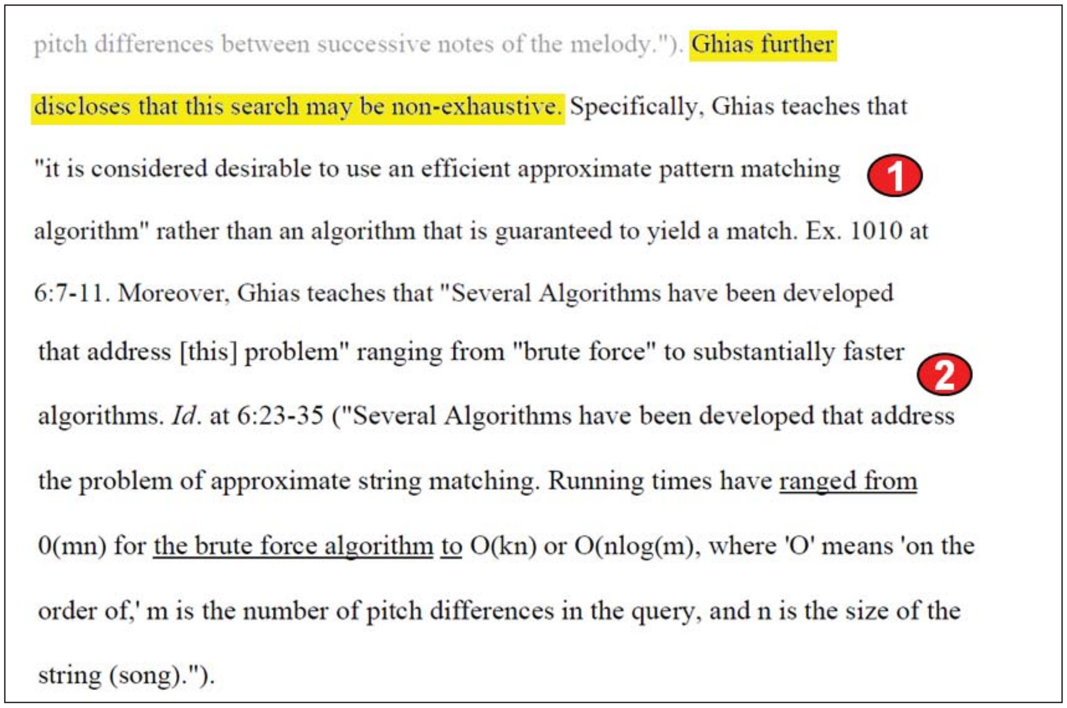
391. Petition Chart: The chart in the Petition cites to these same two passages from Ghias:



'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 "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: 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 ranged from 0(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.

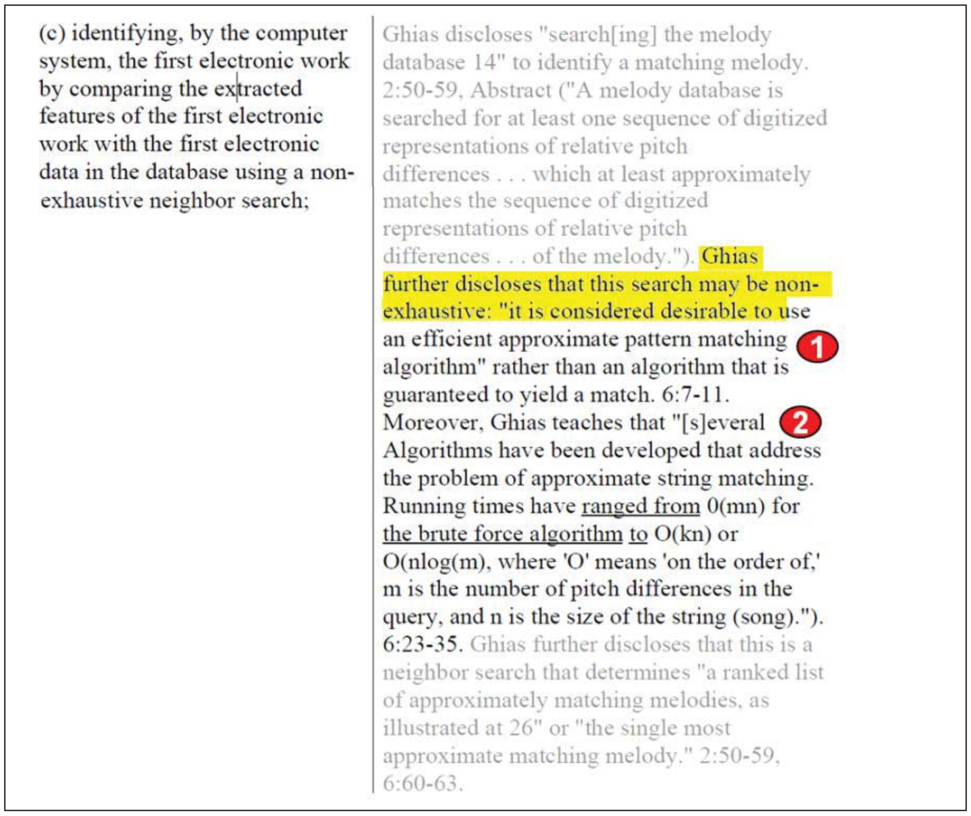
Pet. ('179) at 51.

392. Declaration: Petitioner's Declarant addresses these same two passages:


pitch differences between successive notes of the melody."). 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 0(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:



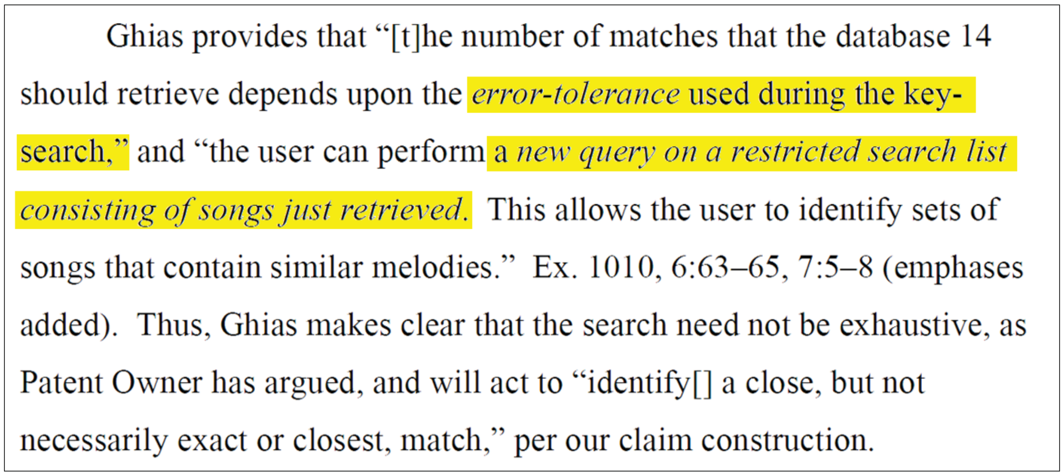
(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Ghias discloses "search[ing] the melody database 14" to identify a matching melody. 2:50-59, Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences . . . 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. 1 Moreover, Ghias teaches that "[s]everal 2 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 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.

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.

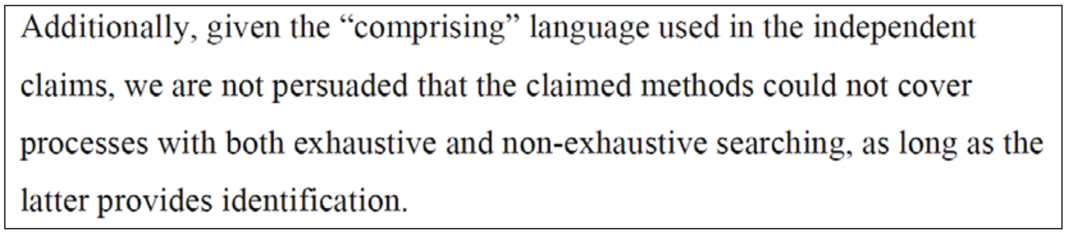
Decision ('179) at 14.

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.

<u>Reason 2</u>: 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).

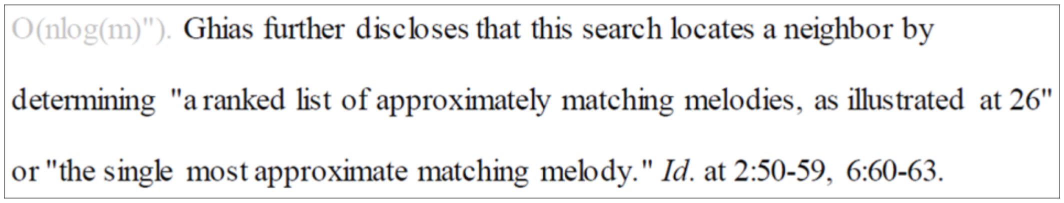
398. I note that in instituting this Ground, the Board did not specifically address whether Ghias discloses the claimed neighbor search. The search disclosed in Ghias is not a neighbor search.

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).

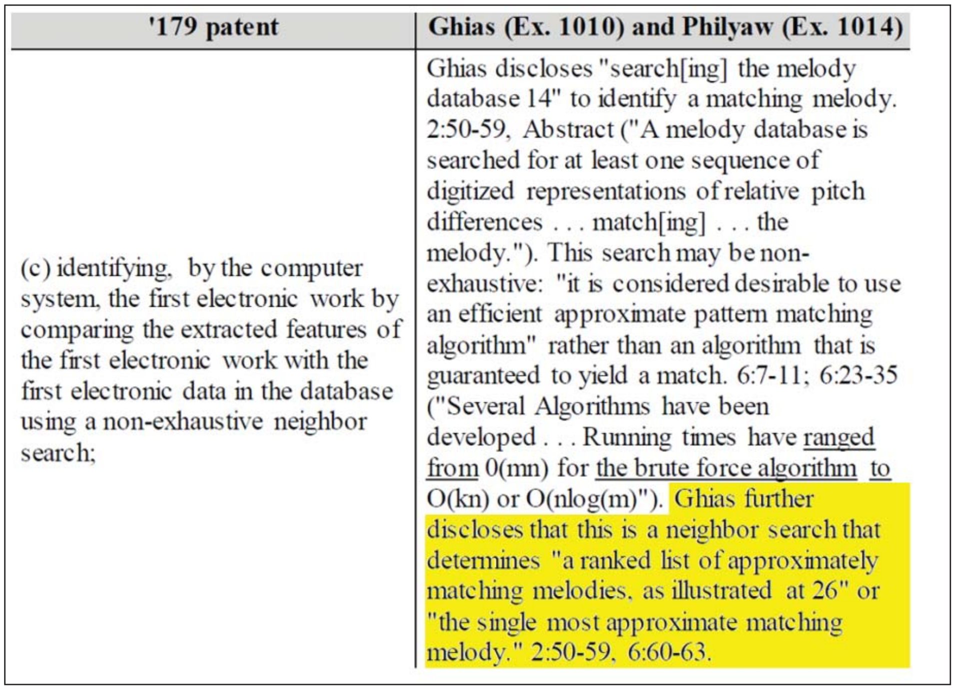
400. The Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a neighbor search.

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


O(nlog(m)"). 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.

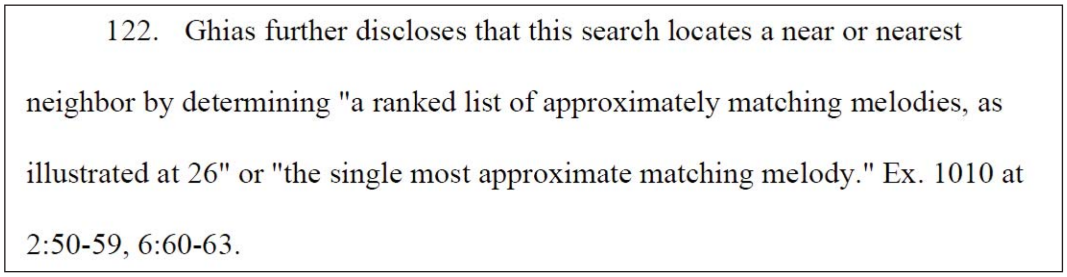
402. The chart in the Petition presents the same reason and quotes the same passages from Ghias:



'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 0(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.

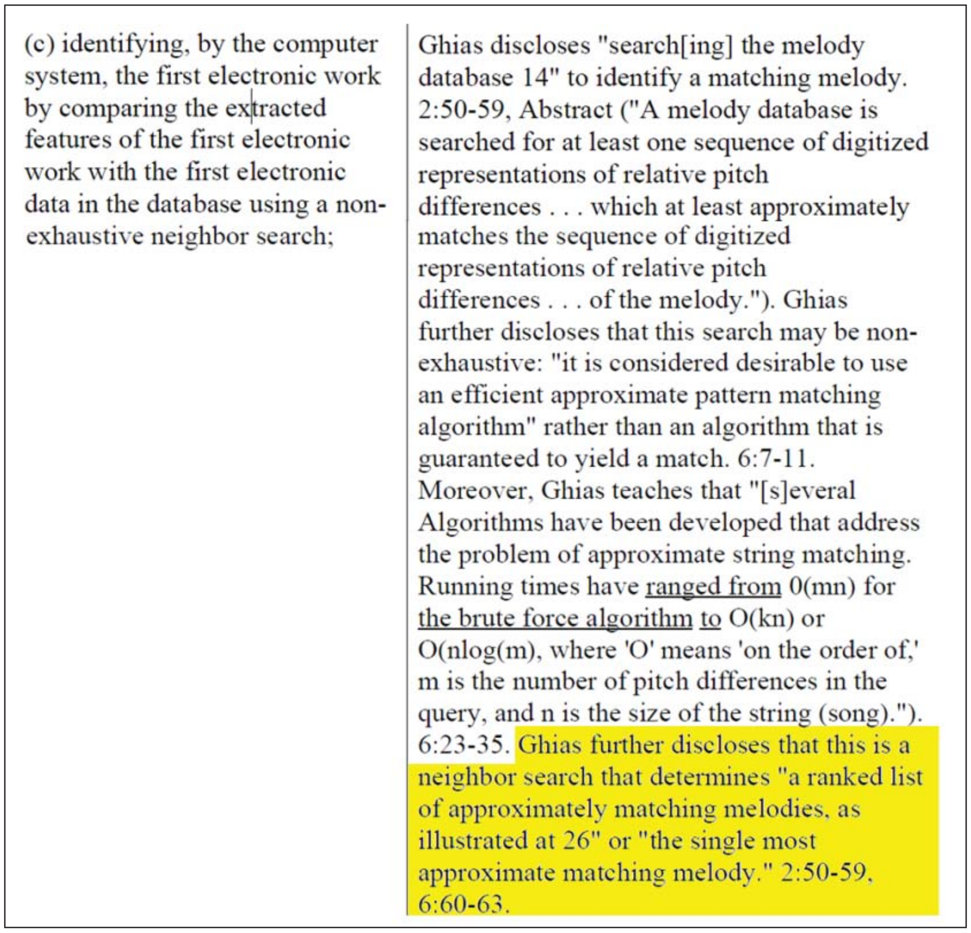
403. The Petitioner’s Declarant also presents the same reason and cites the same two passages from Ghias:



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:

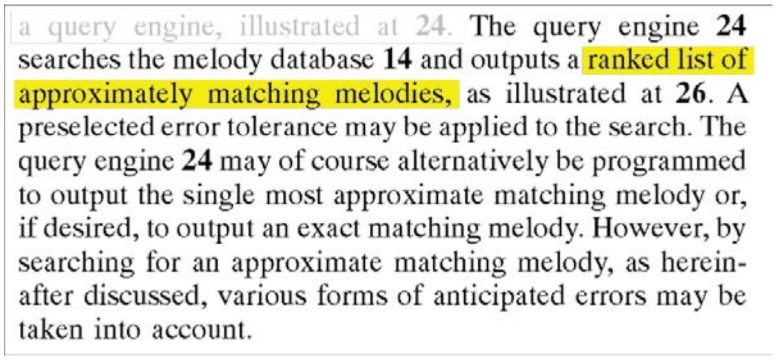


(c) identifying, by the computer system, the first electronic work by comparing the extracted features of the first electronic work with the first electronic data in the database using a non-exhaustive neighbor search;	Ghias discloses "search[ing] the melody database 14" to identify a matching melody. 2:50-59, Abstract ("A melody database is searched for at least one sequence of digitized representations of relative pitch differences . . . 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> 0(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.

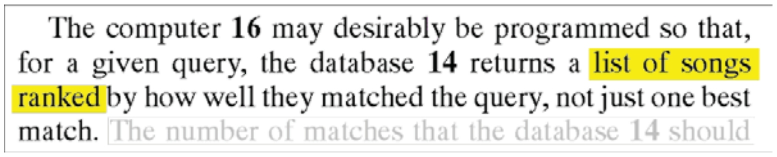
405. Neither of the cited passages from Ghias discloses the claimed neighbor search because, as described above, searches that produce either the ranked list or single most approximate matching melody always identify the closet match. I address each passage in turn.

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


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.

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. The number of matches that the database 14 should

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.”

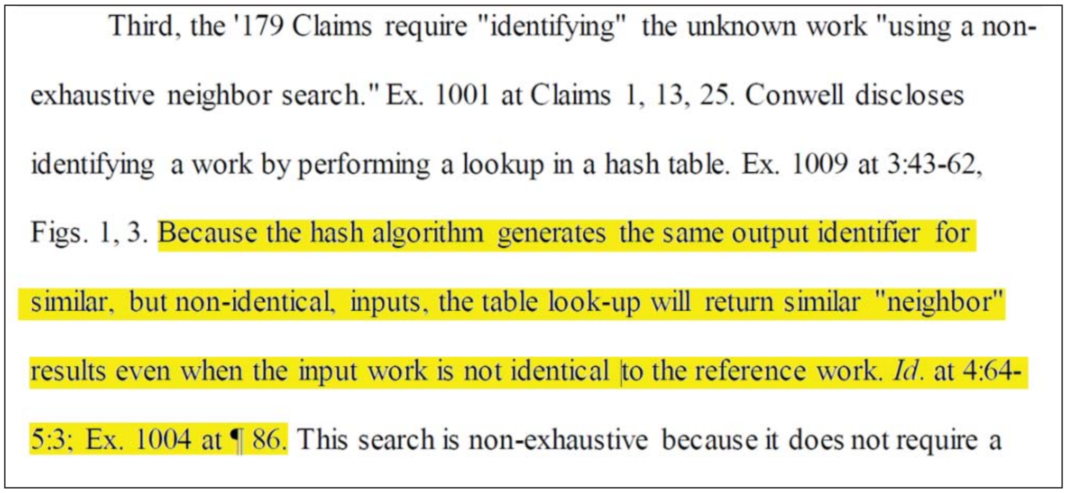
412. As I demonstrated above with respect to the ‘179 Patent,

- 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

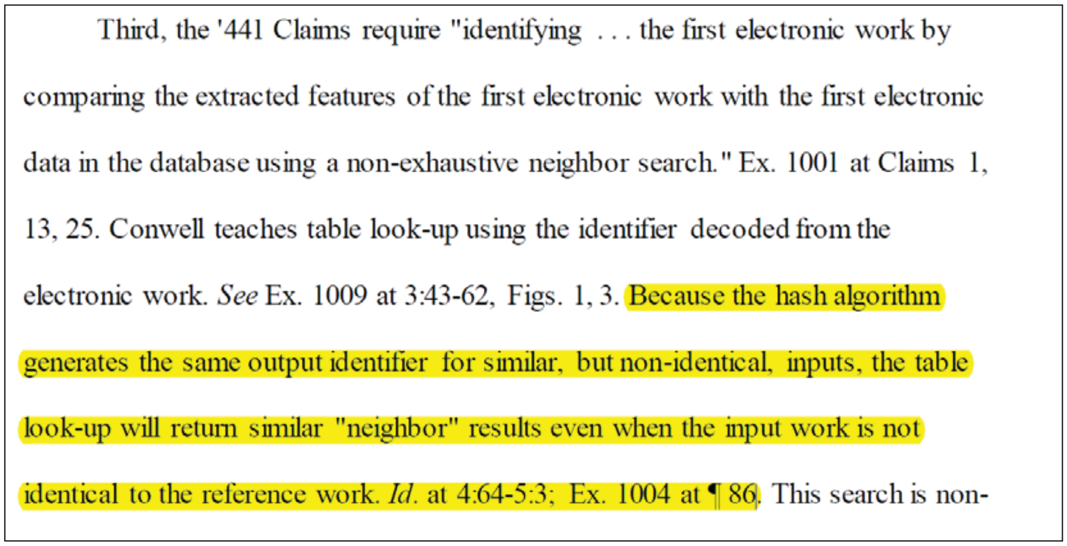
discloses the claimed "neighbor search" for the '441 Patent, the Petition and corresponding Declaration rely on the same discussion and citations to Conwell that they identified for this claim element with respect to the '179 Patent addressed above. Compare Pet. ('179) at 24 with Pet. ('441) at 23-24;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

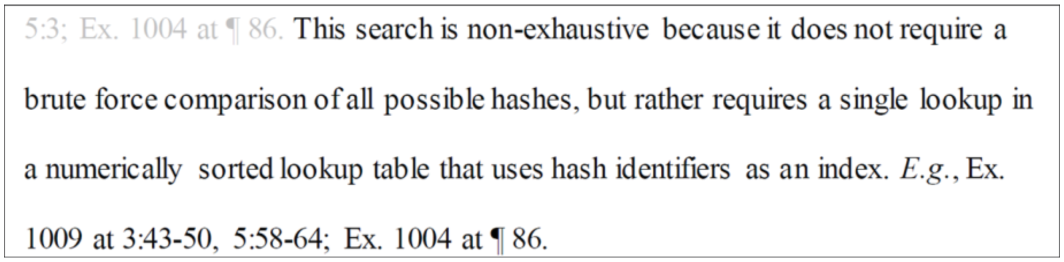
(1) the claimed “non-exhaustive … search” is a search that “locates a match without a comparison of all possible matches,” and

(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.

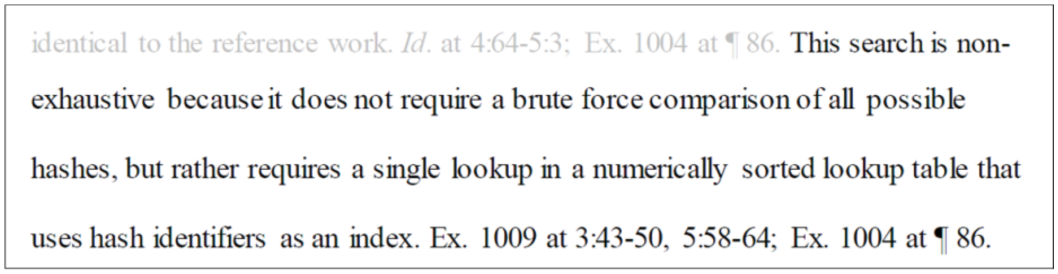
Compare Pet. (‘179) at 24 with Pet. (‘441) at 23-24; Moulin Decl. (‘179) ¶86 with Moulin Decl. (‘441) ¶86.

418. For the ‘179 Patent, Petitioner asserts:


5:3; Ex. 1004 at ¶ 86. 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:


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 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).

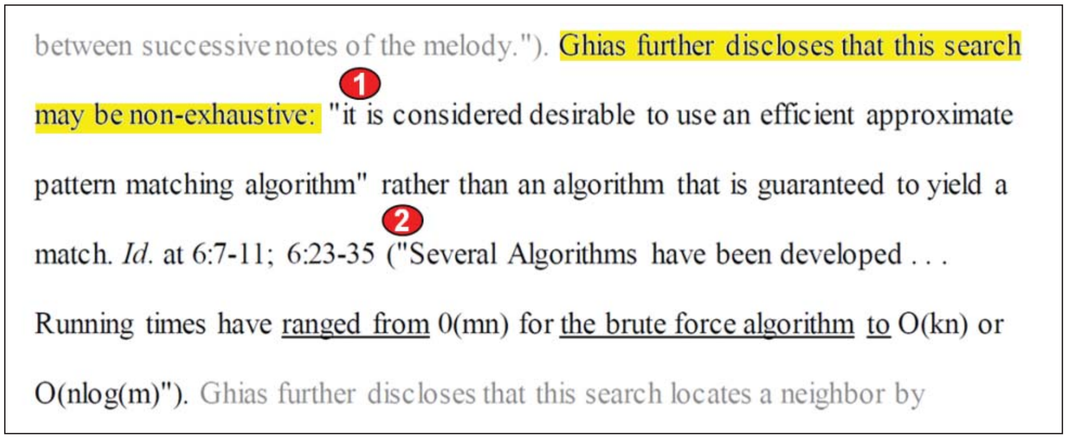
424. As I explained above in detail:

(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.

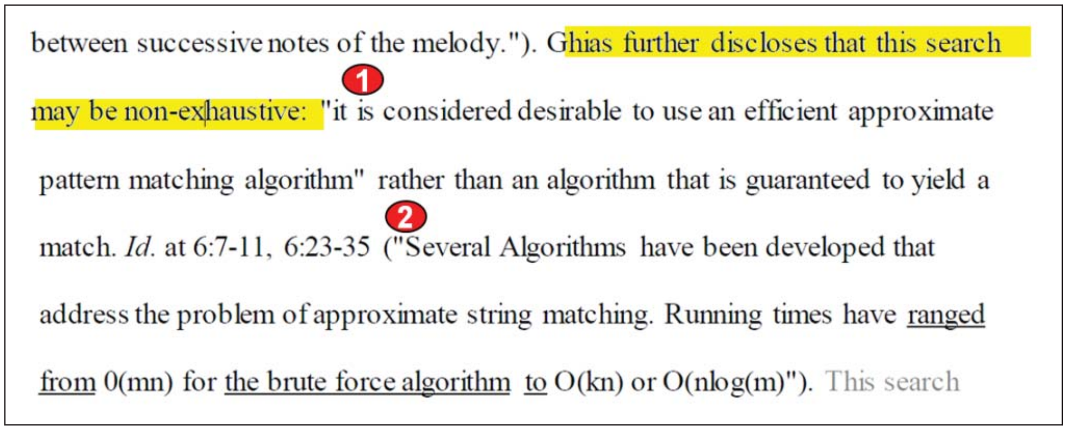
426. For the '179 patent, Petitioner asserts:



between successive notes of the melody."). Ghias further discloses that this search 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; 2 6:23-35 ("Several Algorithms have been developed . . . Running times have ranged from 0(mn) for the brute force algorithm to O(kn) or O(nlog(m)"). Ghias further discloses that this search locates a neighbor by

Pet. ('179) 47-48.

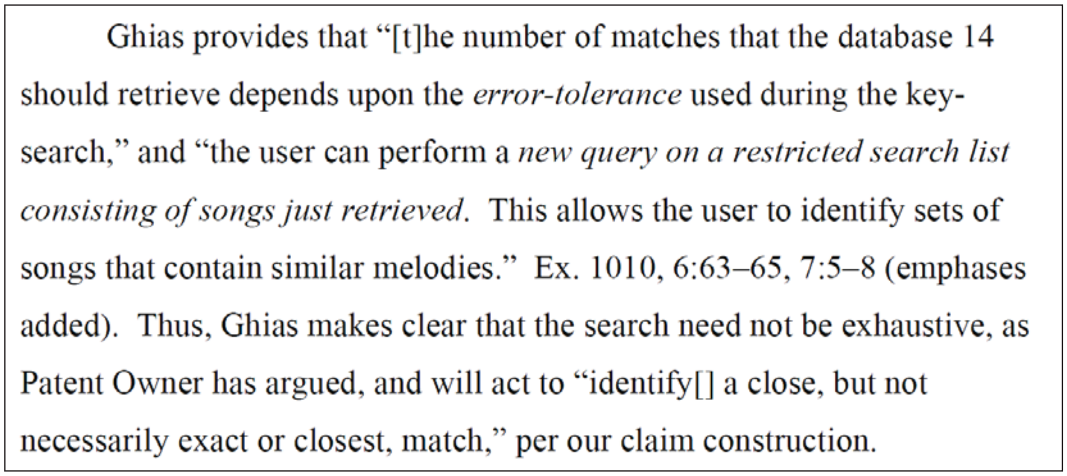
427. For the '441 Patent, Petitioner makes the same assertion:



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 <u>ranged from</u> 0(mn) for <u>the brute force algorithm</u> <u>to</u> O(kn) or O(nlog(m)"). This search

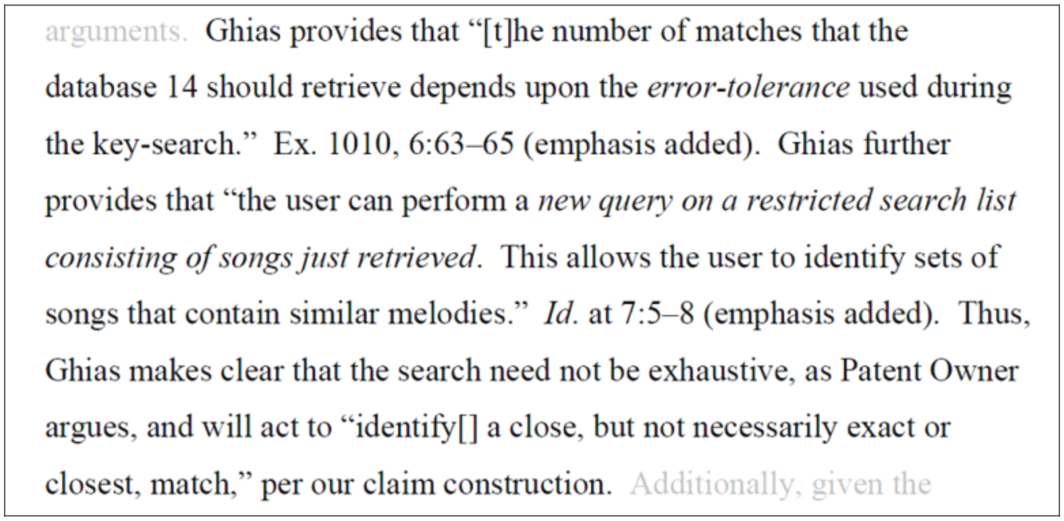
Pet. (‘441) 47-48. Accordingly, the Petition, Declaration, and corresponding charts fail to demonstrate that Ghias teaches comparing the extracted features using a non-exhaustive search for the same reasons set forth above in detail with respect the ‘179 Patent.

428. <u>*Board concerns*</u>: 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.


arguments. 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. Additionally, given the

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

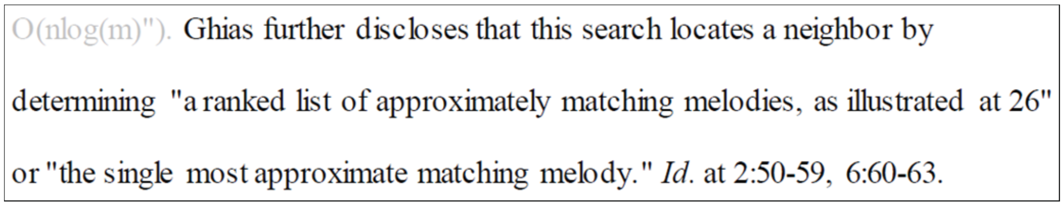
disclose a neighbor search. As explained above in detail with respect to the ‘179 Patent:

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

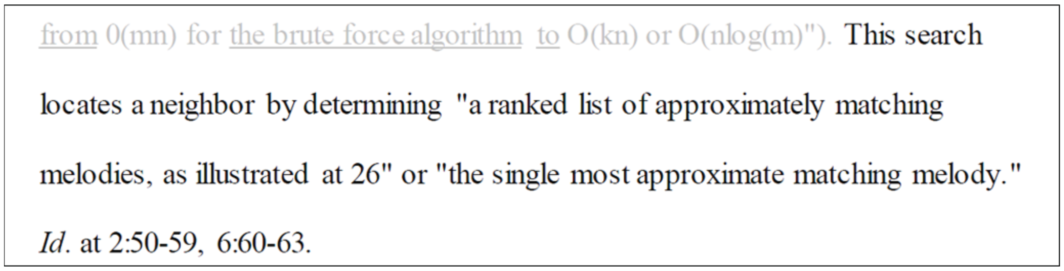
(b) Ghias does not disclose a neighbor search because the disclosed search always (necessarily) identifies an exact or the closest match.

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:


O(nlog(m)"). 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:


from 0(mn) for the brute force algorithm to O(kn) or O(nlog(m)"). 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 [illegible]

Exhibit A

George Karypis
Professor

Department of Computer Science & Engineering
4-192 EE/CS
200 Union Street SE
Minneapolis, MN 55455

phone: (612) 626-7524
fax: (612) 626-1597
email: karypis@cs.umn.edu
URL: 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. Benjumin/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 Bremyer, 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. *"Milti-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. "Genome 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, solver 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.

<u>Workshop Program Committee Memberships</u>

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.

<u>Reviewer</u>

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.