**2. The Board properly rejected Petitioner's assertion that a "non-exhaustive search" should be construed as "a search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches."**

83. The "all data" clause (that I underlined above) in Petitioner's proposed construction (Pet. ('237) at 5; Decision ('237) at 5-7) would improperly include as a "non-exhaustive" search any search that did not compare "all data" in each record, even if the search were a brute force comparison of each record in the database. As an illustrative example, assume the work to be identified "ABC" is compared with all records in a library, including record "DEF." When comparing "ABC" with "DEF," the algorithm determines that there is no match between "ABC" and "DEF" after just comparing the first letter of the work "A" with the first letter of the record "D." If the algorithm does not unnecessarily compare the second and third letters, then according to Petitioner, the search is not "exhaustive" even though every record is compared.

84. Petitioner's Declarant states that a non-exhaustive search is any search that is not a brute force search, and a "'brute force' search, in turn, is a search wherein a query is compared to every single portion of every single item in a database." Moulin Decl. ('237) ¶43. Petitioner's Declarant, however, provides no

analysis or support for this conclusory assertion which, I understand, is insufficient to satisfy Petitioner's burden in these IPR proceedings.

85. One skilled in the art would understand that the "all data" clause is improper because it is:

- inconsistent with how the non-exhaustive search concept is used in the IPR Patents which describes a linear exhaustive search as one where the search compares the work to all "N entries," not all data within all "N entries" (*see e.g.*, '179, 21:10-42; 8:59-9:54); and
- not part of the ordinary meaning of "non-exhaustive search" (*see* Ex. 2001).

86. Moreover, objective sources confirm my understanding that an "exhaustive" or "brute-force" search systematically compares the work with each record in a database, not all data within each record, for example:

> "In computer science, brute-force search or <u>exhaustive search</u>, also known as generate and test, is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and <u>checking whether each candidate satisfies the problem's statement</u>."

Ex. 2001—each "candidate" is checked, not "all data" within each candidate.

87. Petitioner's own Declarant twice confirmed my understanding—that a "non-exhaustive" search searches a subset of "potential matches," not a subset of "all data within all potential matches":

(1) "Because neighbor searching is computationally intensive, content recognition schemes typically employed search algorithms that increased efficiency by intelligently searching only a subset of potential matches (*i.e.*, 'non-exhaustive' algorithms)." Moulin Decl. ('237) ¶12;

(2) "to maximize search efficiency, persons skilled in the art routinely employed more efficient searches that did not conduct a comparison of every single item in a database, sometimes referred to as non-exhaustive searches." Moulin Decl. ('237) ¶43.

88. For the reasons that I presented above, one skilled in the art would understand that the Board properly rejected Petitioner's "all data" clause. Decision ('237) at 6.

### C. neighbor search / identifying a neighbor / neighbor / near neighbor ('237, '988, '179, and '441 patents).

89. One skilled in the art would understand that the Board properly construed a "neighbor search" and "identifying a neighbor" as "identifying a close, but not necessarily exact or closest, match" and "neighbor" and "near neighbor" as

"a close, but not necessarily exact or closest, match." Decision ('237) at 8; Decision ('988) at 7-8; Decision ('179) at 8; Decision ('441) at 7.

90. Petitioner and its Declarant agree with the Board's construction of "neighbor search." *See e.g.*, Petition ('179) at 6 ("The term 'neighbor search' … should be construed to mean 'identifying a close, but not necessarily exact, match.'"); Moulin Decl. ('179) ¶45 ("'neighbor search' means 'identifying a close, but not necessarily exact, match.'"); Moulin Depo. 250:2-5.

91. One skilled in the art would understand that there are two relevant features of a neighbor search under this construction:

92. <u>Feature 1</u>: If a search necessarily identifies an exact or the closest match (*i.e.*, the search is designed to guarantee that an exact or the closest march is identified each time the search is performed), it is not a neighbor or near neighbor search because it is not a search that "identif[ies] a close, but not necessarily exact or closest, match." Rather, such a search necessarily identifies an exact or the closest match.

93. <u>Feature 2</u>: If a search that necessarily identifies an exact or the closest match (*e.g.*, Match 1) but also identifies other matches that, by definition, are not the closest match (Match 2, Match 3, Match 4), the search still necessarily identifies an exact or the closest match (Match 1) and therefore cannot be the claimed neighbor or near neighbor search.

**D. approximate nearest neighbor search (‘237 patent).**

94. As I noted above, the Petitioner did not identify a construction of “approximate nearest neighbor search.”

95. The Board preliminary determined that an “approximate nearest neighbor search” is a search “identifying a close match that is not necessarily the closest match.” Decision (‘237) at 9. One skilled in the art would understand that this construction is correct, but incomplete, as demonstrated by the ‘237 specification. The ‘237 specification states that the claimed “approximate nearest neighbor search” is [1] a sub-linear neighbor search that [2] does not always find the closest point to the query—*i.e.*, does not always find the closest match:

> “[1] One example of a sub-linear time search is an approximate nearest neighbor search. [2] A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point.”

‘237, 9:12-19.

96. The first feature—that a “approximate nearest neighbor search” is a sub-linear time search—is not reflected in the Board’s preliminary construction and, as demonstrated below, should be included in the construction. The second

feature of the claimed "approximate nearest neighbor search" is reflected in the Board's preliminary construction—"identifying a close match that is not necessarily the closest match." I address these two features in reversed order.

### 1. "identifying a close match that is not necessarily the closest match"

97. This feature of "approximate nearest neighbor search" was properly adopted by the Board. A search that is guaranteed to return the actual closet match is not an "approximate nearest neighbor search." The '237 specification states that an "approximate nearest neighbor search does not always find the closest point to the query." '237, 9:15–16. Accordingly, a search that "always finds" (*i.e.*, is guaranteed to find) the closest match is not an "approximate nearest neighbor search" while a search that is not guaranteed to find the closest match can be an "approximate nearest neighbor search" if it identifies a close match. *See* Pet. ('237) at 19 (stating that a reference discloses an "approximate nearest neighbor search" because the search "identifies a neighbor, but not necessarily the nearest neighbor.").

98. This understanding of "approximate nearest neighbor search" is consistent with the ordinary meaning of the phrase

> "<u>Approximate nearest neighbor</u> In some applications it may be acceptable to retrieve a 'good guess' of the nearest neighbor. In those

> cases, we can use an algorithm which doesn't guarantee to return the actual nearest neighbor in every case, in return for improved speed or memory savings."

Ex. 2008 (http://en.wikipedia.org/wiki/Nearest_neighbor_search#Approximate_nearest_neighbor.) at 5.

99. Similar to the neighbor and near neighbor searches addressed above, one skilled in the art would understand that a search that necessarily identifies both: (1) an exact match or the closest match, and, in addition, (2) "a close match that is not necessarily the closest match" is not an "approximate nearest neighbor search" because it is always guaranteed to identify the closest match.

**2. "sublinear"**

100. It is my understanding that an inventor may act as his or her own lexicographer in defining terms used in a patent's claims. One skilled in the art would understand that the '237 patent defines "approximate nearest neighbor search" as a type of sub-linear search.

101. Title: In the title of the '237 patent, the patentee identified an "approximate nearest neighbor search" as a type of sub-linear search: "Identifying works, using a sub-linear time search, such as an approximate nearest neighbor search, for initiating a work-based action, such as an action on the internet." '237, Title.

102. Abstract: It is my understanding that the abstract of a patent may be used to determine the scope of the invention. In its Abstract, the '237 patent also describes an "approximate nearest neighbor search" as a "sub-linear time search": "determining an identification of the media work . . . using a sub-linear time search, such as an approximate nearest neighbor search for example." '237, Abstract.

103. Specification: In describing methods for carrying out a sub-linear search of the reference data set, the '237 specification also describes an "approximate nearest neighbor search" as a type of sub-linear search: "One example of a sub-linear time search is an approximate nearest neighbor search." '237, 9:12–14.

104. In its preliminary construction, the Board did not include the sublinear feature of the claimed "approximate nearest neighbor search" based on what appears to be faulty logic. The Board preliminarily found:

> We largely agree with Patent Owner's construction, but note that the Specification refers to "[o]ne example of a sub-linear time search is an approximate nearest neighbor search" (Ex. 1001, 9:12–14), such that we are not persuaded that an "approximate nearest neighbor search," must be a sub-linear search, as that term has been construed above. As such, we are persuaded that the proper construction of "approximate nearest neighbor search" is "identifying a close match that is not necessarily the closest match."

Decision ('237) at 9. The logic underlying the Board's reasoning appears to be as follows: If A is "one example" of B, A is not always B. In my opinion, this logic is faulty.

105. If A is "one example" of B, A is always B even though there may be examples other than A that fall within the scope of B. If A is "one example" of B, the scope of B is not limited to just A (*i.e.*, the scope of B can include C, D, and E) but A is always B. For example, a poodle is "one example" of a dog; a poodle is always a dog (there is no scenario where a poodle is not a dog) but there are other examples that fall within the scope of dog beyond poodles, *i.e.*, terriers, Dalmatians, etc. Just like a "poodle" being "one example" of a dog must be a dog (*e.g.*, a dog bred with a curly coat that is usually clipped …) an "approximate nearest neighbor search" being "one example" of a "sublinear search that ….."

must be a sublinear search (*i.e.*, a "sublinear search identifying a close match that is not necessarily the closest match.")

## VI. '237 patent.

106. I understand that the Board instituted the '237 IPR based on three Grounds:

- Ground 1: Claims 1, 3–5, 7–9, 11–13, 15, 16, 21–25, 29, 30, 33, 37, and 38 as unpatentable under 35 U.S.C. § 102(e) as anticipated by Iwamura;
- Ground 2: Claims 1–3, 5–7, 9–11, 13–15, and 21–24 as unpatentable under 35 U.S.C. § 102(b) as anticipated by Ghias; and
- Ground 3: Claims 26, 27, 34, and 35 as unpatentable under 35 U.S.C. § 103 as obvious over Iwamura and Chen.

Decision ('237) at 21-22. I address each Ground in turn.

### A. '237 Ground 1: The instituted claims of the '237 patent are not anticipated by Iwamura.

107. The Board instituted Ground 1 based on the following: Claims 1, 3–5, 7–9, 11–13, 15, 16, 21–25, 29, 30, 33, 37, and 38 as unpatentable under 35 U.S.C. § 102(e) as anticipated by Iwamura. Decision ('237) at 21 (I underlined the

independent claims). Ground 1 fails because Iwamura does not disclose the following key elements from each instituted independent claim:

- sub-linear time search (claims 1, 5);
- approximate nearest neighbor (claims 9 and 13);
- nonexhaustive search … to identify a near neighbor (claim 25); and
- sublinear approximate nearest neighbor search (claim 33).

I address each in turn.

#### 1. sub-linear time search (claims elements 1(b) and 5(b.2)).

108. Claims elements 1(b) and 5(b.2) require a "sub-linear time search."

109. As I explained above, a "sub-linear time search" is "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched." Decision ('237) at 7.

110. One skilled in the art would understand that Iwamura does not disclose a "sub-linear time search." Iwamura discloses a searching algorithm that is designed to be more efficient than alternatives by comparing peak notes from the work to be identified with the peak notes in the database. Iwamura, 6:59-60; 12:1-2. While the individual comparisons of a work to a record in the library can be more efficient using this peak note approach, Iwamura does not teach an algorithm that "scales with a less than linear relationship to the size of the data set to be searched" where the data set is either (a) the number of records in the database, or

(b) even the length of an individual record. Instead, each melody in the melody database is processed as part of the disclosed comparison and "[t]he reference melody that gives the least difference is returned as a search result." Iwamura, 7:53-55.

111. Specifically, Iwamura confirms that the referenced Boyer-Moore algorithm (the basis for alleged disclosure of a sub-linear search in the Petition, Declaration, and Decision) searches all items in the database and even searches "word by word from the beginning of the database to the end" and therefore cannot scale with a less than linear relationship to the size of the data set being search—*i.e.*, it is not sublinear:

> "Boyer Moore (discussed below) or other string-matching algorithms do not have this kind of flexibility. They only search word by word from the beginning of the database to the end."

Iwamura, 9:52-55.[15]

112. The search algorithms disclosed in Iwamura do not reduce the number of records to be searched during a search (or even the data to be searched within a record) as the dataset increases. Rather, the disclosed algorithms speed up the comparison of the work to each record by matching peaks. Iwamura, 9:9-11. Accordingly, the disclosed algorithms in Iwamura search all records in the library,

---

[15] The word-by-word comparison is valid for the worst case.

and the computational time that the disclosed search takes to make such comparisons grows linearly with the number of records in the database (the relevant analysis) and even linearly with the data in each record. Iwamura therefore teaches a linear search rather than the claimed "sublinear" search as the term is used in the IPR Patents, because the computational time that it takes to perform a search grows linearly as new data is added to the database.

113. The Petition fails to satisfy its burden of demonstrating that Iwamura teaches a "sub-linear time search." As support for the "sub-linear" elements, Petitioner (and corresponding Declaration) exclusively relies on the Boyer-Moore algorithm referenced in Iwamura:

114. Petition: The text of the Petition does not address the sub-linear elements or state that Iwamura discloses a "sub-linear time search." Pet. ('237) at 7-10. Neither the word sublinear nor the concept appears in the text of the Petition.

115. Petition Chart: In its chart, Petitioner exclusively relies on the referenced Boyer-Moore algorithm as support for the sub-linear search elements (highlighted in yellow in the passages below):

Claim 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Iwamura determines an identification of the media work using the extracted features by "find[ing] the closest melody from the database," which is a neighbor. 9:25-38, 12:1-2. Iwamura discloses searching using the "Boyer-Moore algorithm" (9:63-64, 10:1-3), which is sublinear (Ex. 1017 at 1). Ex. 1004 at ¶ 72. |

Pet. ('237) at 10-11.

Claim 5(b.2) (Petitioner references Claim 1):

| | |
|---|---|
| 2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and | Petitioner incorporates the above discussion of Iwamura regarding Claim 1b. |

Pet. ('237) at 12.

116. <u>Declaration</u>: The Declaration also exclusively relies on the Boyer-Moore algorithm as support for the sublinear search elements:

> 72. It is my opinion that Iwamura further teaches how this search can be sublinear. For example, Iwamura discloses that different "search algorithms may be applied to perform melody searches," (*id.* at 10:2-3), such as the "Boyer-Moore algorithm," (*id.* at 9:63). "On the average the [Boyer-Moore] algorithm has a sublinear behaviour." Ex. 1017 at 1.

Moulin Decl. ('237) ¶72.

117. <u>Declaration Chart</u>: The chart in the Declaration also exclusively relies on the Boyer-Moore algorithm:

Claim 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Iwamura discloses the use of a "search engine" to determine an identification of the media work using the extracted features by "find[ing] the closest melody from the database," which is a neighbor. 9:25-38, 12:1-2. Iwamura discloses searching using the "Boyer-Moore algorithm" (9:63-64, 10:1-3), which is sublinear (Ex. 1017 at 1). |

Moulin Decl. ('237) ¶75.

Claim 5(b.2) (the Declarant references Claim 1):

| | |
|---|---|
| 2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and | I incorporate my above discussion of Iwamura regarding Claim 1b. |

Moulin Decl. ('237) ¶75.

118. Neither the Petition nor Declaration identifies any basis for asserting that Iwamura discloses the sub-linear search elements other than the referenced Boyer-Moore algorithm. Pet. ('237) at 10-12; Moulin Decl. ('237) ¶72. My understanding is confirmed by Petitioner's Declarant:

```
        Q    The only thing you identify in your
Declaration about Iwamura that could disclose a
sublinear time search is the Boyer-Moore algorithm;
correct?
        A    As far as I remember, yes.  In that
Declaration at that time, yes.
```

Moulin Depo. 82:23-83:3.

119. One skilled in the art would understand that the referenced Boyer-Moore algorithm, however, does not disclose or even address a sublinear search—that is "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched." Decision ('237) at 7. Because Iwamura itself does not state that Boyer-Moore algorithm is sublinear, the entire basis in the Petition and corresponding Declaration for the claimed sublinear elements is the single statement in the Petitioner's Declaration:

"On the average the [Boyer-Moore] algorithm has a sub-liner behavior." Moulin Decl. ('237) ¶72 (quoting Ex. 1017 at 1). One skilled in the art would understand that this statement is not accurate with respect to the relevant sub-linear behavior, *i.e.*, with respect to the size of the database. My understanding was confirmed by Petitioner's Declarant who testified that:

(1) he understood that "sub-linear" in the context of the '237 patent is based on the size of the data set searched, not the size of the query or pattern to be matched (from the work to be identified);

(2) the Boyer-Moore algorithm does not disclose a search that is sublinear with respect to the dataset or database or even the length of a record to be search (it does not even address a database or dataset); and

(3) that when he wrote “which is sublinear” in his Declaration, he did not intend the Board to interpret “sublinear” in the context of the ‘237 patent but instead in a different context unrelated to ‘237 patent.

120. (1) As I noted above, Petitioner’s Declarant understood that “sublinear” in the context of the ‘237 patent is based on the size of the searched dataset, not the size of the query or pattern of the work to be matched (which is the correct understanding):

> 53. I understand and agree with Petitioner's position that the term "sublinear search" means "a search whose execution time has a sublinear relationship to database size." For instance, a linear search of a 200-item database would take twice as long as a linear search of a 100-item database. By contrast, a sublinear search of a 200-item database would take less than twice as long as a sublinear search of a 100-item database, perhaps, for instance, 1.5 times as long.

Moulin Decl. (‘237) ¶53.

Q Is it the case that in doing your analysis to determine whether or not prior art anticipated, you applied the definition that a sublinear search or a sublinear time search was one whose execution time has a sublinear relationship to the database size?
A That is correct, yes.
Q Is that the correct definition that should be applied, or are you applying the wrong definition?
A In my opinion, this is the correct definition to be used.

Moulin Depo. 24:1-12.

Is it the case that for a search to be sublinear as it's used in the '237 patent, it's not enough for it to have execution time that is sublinear in relationship to the size of the pattern; it must also be sublinear in relationship to the size of the database?
A When I read "sublinear" in, say, Claim 5 of the patent, as we just did, I understand sublinear to mean in relation with the size of the database. It does not say anything about in relation with the size of the query.

Moulin Depo. 26:11-21.

Q Would it be wrong, then, to assess whether a search is sublinear by applying a definition that said, "It's sublinear if this execution time has a sublinear relationship to the size of the query or the pattern"?
A It would not be very relevant. Again, mathematically, it can be done. Everything can be done. But it would not be relevant, from an engineering viewpoint, for an application like this.

Moulin Depo. 25:4-12.

Let's assume that the Patent Trial and Appeal Board was presented with prior art that presented a search that was linear with respect to the size of the database, but it was sublinear with respect to the size of the pattern.
Would that prior art demonstrate a sublinear search as it's used in Claim 25?
A People say it's a linear search, again, because it's in relation with the size of the database. And as you just said, that complexity is still linear; so people would say it is a linear search.
Q It is not a sublinear search?
A It's only sublinear in terms of the size of the query, which is generally not the parameter of -- the relevant parameter.

Moulin Depo. 26:25-27:15.

Q Is it the case that if we had a piece of prior art that was linear with respect to the size of the database but sublinear with respect to the size of the query or the pattern, that that prior art would not teach a sublinear search as it's used in Claim 25?
A Again, if one understands sublinear to be in terms in relation to the size of the database, that would be a -- a linear search.

Moulin Depo. 27:16-24.

Let's assume we've got a piece of prior art that scales at a sublinear relationship with the size of the pattern or query but it scales at a linear relationship with the size of the database that's being searched.
Would that prior art demonstrate or disclose a sublinear search as it's used in the claims of the '237 patent?
A No. Again, because my understanding is the claims of the '237 patent, whenever there's mention of "sublinear," it means in terms of the database size. It does not say it explicitly; it's my inference based on my knowledge and my expertise.

Moulin Depo. 28:4-16.

> Q When you wrote your Declaration, you had in mind that sublinear meant having execution time that increased at a less-than-linear relationship compared to the size of the dataset being searched; right?
>
> A That's in the context of database search, yes.
>
> Q And then you wrote this claim chart in order to indicate where in Iwamura it disclosed each part of the claim; right?
>
> A That's correct, yes.

Moulin Depo. 77:14-24.

121. (2) Petitioner's Declarant confirmed my understanding—that the Boyer-Moore algorithm referenced in Iwamura does not disclose a search that is sublinear with respect to the database size (*i.e.*, the size of the data set to be searched)—it does not even address a database (Moulin Depo. 53:19-22 ("There's no database in Boyer-Moore."))—but instead has a relationship to the size of the query pattern from the work to be identified:

> Q Are you familiar with any analysis of the Boyer-Moore algorithm with respect to the size of the dataset being searched?
>
> A It's described here. So, again, this i, if you look at the worst case, i is N minus patlen, then you obtain it. As I said, it will be a linear relationship.

Moulin Depo. 61:18-24; 44:20-46:6; 59:6-9; 61:25-62:9; 68:25-69:4.

122. (3) Petitioner's Declarant confirmed my understanding—that the statement in his Declaration—Petitioner's only support for the sub-linear elements—was wrong. He testified that when he wrote:

> algorithm," (*id.* at 9:63). "On the average the [Boyer-Moore] algorithm has a sublinear behaviour." Ex. 1017 at 1.

(Moulin Decl. ('237) ¶72) and wrote just a few pages earlier:

> 53. I understand and agree with Petitioner's position that the term "sublinear search" means "a search whose execution time has a sublinear relationship to database size." For instance, a linear search of a 200-item database

(Moulin Decl. ('237) ¶53), he was not trying to convey that the Boyer-Moore algorithm was sublinear or "has a sublinear behavior" in the context of the '237 patent –*i.e.*, "has a sublinear relationship to the database size":

> Q When you wrote this, were you trying to convey that searching using the Boyer-Moore algorithm would be sublinear with respect to the size of the dataset being searched?
>
> A No. I did not -- no.

Moulin Depo. 74:20-24; 74:8-12.

> Q When we read paragraph 72, are you conveying to the reader that the Boyer-Moore algorithm is sublinear with respect to the size of the dataset being searched?
>
> A No. All I do is quote a part of a paper that shows why that algorithm is much faster than brute force. That's all I'm doing. You're inferring things I'm not saying or writing.

Moulin Depo. 69:9-16.

> In paragraph 72, are you representing to the Board that it's your understanding that the Boyer-Moore algorithm has a sublinear behavior with respect to the size of the dataset?
>
> A No. This is just a quote of another paper discussing Boyer-Moore. This Boyer-Moore algorithm has been used in a variety of contexts, including, of course, content recognition. I'm simply quoting from another paper here. I'm not presenting anything about what you asked.

Moulin Depo. 66:9-18.

> When you wrote this sentence here on page 29, did you think that someone at the Board looking at this might think that you meant that a -- the Boyer-Moore algorithm was sublinear as used in the patent claim?
>
> A I didn't think of it that way, no.

Moulin Depo. 75:23-76:3.

Q And when you wrote this, were you trying
to convey to the Board that the Boyer-Moore
algorithm is one that, if you use it, it's sublinear
with respect to the size of the database?
A No. No.

Moulin Depo. 67:17-21.

123. Consistent with my understanding, Petitioner's Declarant clarified that he was not claiming that the Boyer-Moore algorithm referenced in Iwamura discloses a sub-linear search in the context of the '237 patent, *i.e.*, with respect to the size of the dataset:

Q And next to the phrase "a sublinear time
search" in the claim, you wrote, "The Boyer-Moore
algorithm, which is sublinear"; right?
A It's not me who wrote it. I'm quoting
from a reference.
Q Well, you wrote the words "which is
sublinear"; right?
A I quoted from a reference, again, showing
why there are much faster alternatives to brute
force.
Q Let me rephrase it --
A I'm not -- again, to make it very clear,
I'm not claiming that using Boyer-Moore simply alone
is going to yield a sublinear search for that
database problem. I'm not claiming that. Just to
be -- to make it clear.

Moulin Depo. 77:25-78:15.

Q Okay. Would you agree, sir, that if -- that one way to read this would be that you were claiming that the claim language, "perform a sublinear time search," was satisfied by searching using the Boyer-Moore algorithm?

A That might be one way of reading it. It's not the way I'm reading this now.

As I said, the way I'm reading this is I'm quoting language from a reference. And, again, to make the record clear, I'm not claiming that Boyer-Moore, as this alone, is going to give us a sublinear time search in a database search problem. I'm not claiming that. And I did not claim it in this document.

Moulin Depo. 78:16-79:6.

Q Would it be reasonable for the Board to have read this as you opining, you asserting, that Iwamura discloses a sublinear time search because it discloses searching using the Boyer-Moore algorithm, which is sublinear?

A Again, I don't know how different people can read it. If there's any ambiguity, I hope I just cleared it up. I'm not claiming that Boyer-Moore alone is going to give us a sublinear time search for the database search problem.

Moulin Depo. 79:9-18

Q To be candid with the Board, wouldn't it
have been better to say, "Board, the Boyer-Moore
algorithm is linear, not sublinear"?
A Listen, there are many words that I -- I'm
sure I could have chosen better words. So I agree
with you, there is probably better ways to write
this. I don't dispute that.
Q Do you agree that it would have been
better to tell the Board, "The Boyer-Moore algorithm
is linear, not sublinear"?
A I don't -- I'm not saying it's linear
either. I -- I don't know. All I'm saying is I'm
not representing that the Boyer-Moore algorithm
alone is going to give us a sublinear time search
for database searching. That's all I'm saying.
Q Would the Boyer-Moore algorithm alone give
you a sublinear time search for searching a string?
A Again, are you looking at worst case? The
answer is no.

Moulin Depo. 79:19-80:12; 80:15-83:3.

124. Accordingly, one skilled in the art would understand that the support in the Petition and Declaration for the sublinear search elements fails to disclose the sublinear search elements.

125. *Board's concerns*: I now address the Board's specific concerns (identified in its Decision) with respect to whether Iwamura discloses the claimed "sub-linear time search." In instituting Ground 1, I note that the Board preliminary found that Iwamura disclosed the "sub-linear time search because (a) a sub-linear

search of the data within the records can be sublinear even if every record in the database is searched, and (b) Patent Owner's argument that Boyer-Moore searches all items in the database therefore does not demonstrate that the Boyer-Moore algorithm is not sub-linear:

> the inquiry. In addition, we note that no claim in the '237 Patent requires the searching, in the determining aspect of the claims, to be both nonexhaustive and sub-linear, such that a sub-linear search of the data within the records, even if every record is searched, can potentially teach the aspect of independent claims 1 and 5 which recite "perform[ing] a sub-linear time search of extracted features." Although the Specification of the '237 Patent discloses that a sub-linear search is performed on the records of the database and not information within the records, the claims do not specify that the sub-linear search must be performed on a subset of all of the records, and not information within individual records.

Decision ('237) at 11.

> Patent Owner also argues that Iwamura's use of the "'Boyer-Moore algorithm' searches all items in the database and therefore is not sublinear." Prelim. Resp. 18–19. As discussed above, we are not persuaded that this is a deficiency with respect to the instant claims. With respect to "approximate

Decision ('237) at 12. It is my opinion that the Board's preliminary analysis is flawed on multiple levels for the reasons I explain below.

126. First, the Board's preliminary analysis is based on an incorrect interpretation of the construction of sub-linear as it would be understood by one skilled in the relevant art at the time of the inventions. The Board construed a

"sublinear" search as "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched," not the length of any specific record in the database. As I explained above in detail above and reflected in in the Board's analysis of the construction of sub-linear, the data set is the number of records in the database to be searched—"the size of the data set ("N")." Decision ('237) at 7.

127. In addition, as I explained above in detail, those skilled in the art understand that the size of the data set in the context of the '237 patent refers to the number of records in the database to be searched (N) and not the length of any particular record in the database. This understanding is consistent with Dr. Moulin's explanation in his Declaration. *See* Moulin Decl. ('237) ¶53. Accordingly, the Board's preliminary analysis is based on an improper interpretation of the construction of "sublinear."

128. Second, it is my understanding that the Board's preliminary analysis has the relevant burden backwards—it is not the Patent Owner's burden to demonstrate that the referenced Boyer-Moore algorithm does not disclose a sublinear search. Rather it is my understanding that it was the Petitioner's burden to demonstrate that referenced Boyer-Moore algorithm discloses a sublinear search. As I showed above, Petitioner failed to satisfy this burden. As I explained

above, in my opinion, one of ordinary skill in the art would understand that Boyer-Moore algorithm is not a sublinear search in the context of the '237 patent.

129. Third, one skilled in the art would understand that there is no evidence under any interpretation of sublinear in the context of the '237 patent that the referenced Boyer-Moore algorithm discloses a search that is sublinear with respect to either (a) the "size of the dataset" (Decision ('237) at 7); or (b) the length of an individual record being searched. In my opinion, one of ordinary skill in the art would understand that it is not.

130. The two references to the Boyer-Moore algorithm in Iwamura are:

> omission also contributes to performing a fast search. Boyer-Moore (discussed below) or other string-matching algorithms do not have this kind of flexibility. They only search word by word from the beginning of the database to the end.

Iwamura, 9:52-55.

> however, a key-shift is required for a melody search. There are many studies for fast and efficient string search techniques. For example, the Boyer-Moore algorithm is well-known as one of the best solutions. See

Iwamura, 9:61-64. While the Boyer-Moore algorithm is described as being "efficient," one skilled in the art would understand that neither passage states that the algorithm is sublinear with respect to either the number of references in the database or the length of an individual record to be searched.

131. Fourth, as I explained above, Petitioner's Declarant confirmed my understanding—that the referenced Boyer-Moore algorithm does not disclose a search that is sublinear in the context of the '237 patent.

#### 2. approximate nearest neighbor search (claim elements 9(b) and 13(b.2)).

132. As I presented above, one of ordinary skill in the art would understand that, in the context of the '237 patent, an "approximate nearest neighbor search" is a sub-linear search identifying a close match that is not necessarily the closest match. Also, as I explained above, a search that necessarily identifies the closest match is not an "approximate nearest neighbor search" even if it also identifies other near matches.

133. One skilled in the art would understand that Iwamura does not disclose the claimed "approximate nearest neighbor search" for two independent reasons.

134. Reason 1: One skilled in the art would understand that Iwamura does not disclose an "approximate nearest neighbor search" because Iwamura does not disclose "identifying a close match that is not necessarily the closest match." Iwamura discloses a search that always identifies an exact or the closest match. Consistent with my understanding, Petitioner's Declarant likewise confirmed that

Iwamura will either produce an "exact match" if it finds one, or the "best match it finds using that approximate criterion." Moulin Depo. 271:22-272:12.

135. The system in Iwamura will always find the closest match, even if unimportant peaks are skipped or repeated patterns are avoided. My understanding is consistent with the understanding of Petitioner's Declarant:

- "[W]'re still going to be identifying the closest match" even when "the unimportant peaks are skipped…. Dropping an unimportant part is not going to affect the ability to find the best match." Moulin Depo. 317:14-23.
- "If we implement that feature of Iwamura… skipping a repeated pattern…. It will not affect the ability to find the best match." Moulin Depo. 318:11-18.

136. Petitioner asserts that Iwamura identifies a neighbor because: "the 'search engine will find the closest melody from the database." Pet. ('237) at 8 (*quoting* Iwamura, 9:24-25)); Moulin Decl. ('237) ¶69. A person of ordinary skill in the art would understand that these statements do not disclose an "approximate nearest neighbor search" which is a search identifying a close match that is not necessarily the closest match. Instead, these statements 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

"appropriate nearest neighbor search." *See* '237, 9:15–16 (an "approximate nearest neighbor search does not always find the closest point to the query.").

137. Because the searches disclosed in Iwamura necessarily return the closest match, they are not search algorithms that identify a match that is not necessarily the closest match, as the properly construed claim element requires. Accordingly, in my opinion, Iwamura neither expressly nor inherently (necessarily) discloses an "approximate nearest neighbor search"—a search that does not necessarily find the closest match.

138. Reason 2: One skilled in the art would understand that Iwamura does not disclose an "approximate nearest neighbor search" because Iwamura does not disclose a sublinear search. As I demonstrated above, an "approximate nearest neighbor search" is "one example" of a sublinear search. Also, as I demonstrated above, Iwamura does not disclose a sublinear search. Accordingly, Iwamura does not disclose the claimed "approximate nearest neighbor search."

139. One skilled in the art would understand that the Petition, Declaration, and corresponding charts fail to demonstrate that Iwamura discloses the claimed "approximate nearest neighbor search." As support for the claimed "approximate nearest neighbor search," the Petition and corresponding Declaration rely on (1) the fault tolerance feature, and (2) skipped portions feature, described in Iwamura.

140. Petition: The text of the Petition does not address the claimed "approximate nearest neighbor search"—I note that the words "approximate nearest neighbor search" do not appear in the text of the Petition.

141. Petition Chart: Petitioner provides the following in its claim chart:

Claim 9(b):

| | |
|---|---|
| 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 Iwamura regarding Claim 1b. Furthermore, Iwamura uses an approximate nearest neighbor "search engine [that] has . . . input fault tolerance capability" (10:17-18), and skips "portions that should not be searched" (12:6-7), such as "repeated patterns" (9:36-44), and "unimportant portion[s]" of the melody (9:44-45). |

Pet. ('237) 12.

Claim 13(b.2) (referencing claim element 9(b)):

| | |
|---|---|
| 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 Iwamura regarding Claim 9b. |

Pet. ('237) 13.

142. Declaration: The text of the Declaration also does not address the claimed "approximate nearest neighbor search."

143. Declaration Chart: Petitioner' Declarant provides the following in its claim chart:

Claim 9(b):

| | |
|---|---|
| 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 Iwamura regarding Claim 1b. Furthermore, Iwamura discloses using an approximate nearest neighbor "search engine [that] has . . . input fault tolerance capability" (10:17-18), and skips "portions that should not be searched" (12:6-7), such as "repeated patterns" (9:36-44), and "unimportant portion[s]" of the melody (9:44-45). |

Moulin Decl. ('237) ¶75.

Claim 13(b.2) (referencing claim element 9(b)):

| | |
|---|---|
| 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 Iwamura regarding Claim 9b. |

Moulin Decl. ('237) ¶75.

144. I note that these statements in the Petition (and Declaration) and corresponding passages from Iwamura do not:

(a) provide a construction of "approximate nearest neighbor search,"

(b) explain how Iwamura discloses the claimed "approximate nearest neighbor search,"

(c) explain why the fault tolerance capability and skipped portion are relevant to or disclose an "approximate nearest neighbor search," and

(d) establish that Iwamura discloses an "approximate nearest neighbor search."

145. One skilled in the art would understand that the quoted passages do not disclose an "approximate nearest neighbor search" because the quoted passages do not disclose a search that (a) is not guaranteed to identify the closest match, and (b) is sublinear.

146. First, as I noted above, the passage from element 1(b) cross-referenced in Petitioner's chart ("Petitioner incorporates the above discussion of Iwamura regarding Claim 1b") does not disclose an "approximate nearest neighbor search." As I explained above, one skilled in the art would understand that an "approximate nearest neighbor search" identifies a close match that is not necessarily the closest match. *See* Decision ('237) at 9. The passage cited in the Petition (and corresponding Declaration) confirms that the search disclosed in Iwamura finds "the closest melody from the database." Pet. ('237) at 8 (*quoting* Iwamura, 9:24-35).

147. Second, one skilled in the art would understand that Petitioner's references to searches that have (a) an "input fault tolerance" (Pet. ('237) at 12, *quoting* Iwamura, 10:17-18), or (b) skipped "portions that should not be searched" (Pet. ('237) at 12 *quoting* Iwamura, 12:6-7, 9:36-44, and 9:44-45) do not expressly or inherently (necessarily) disclose a search that does not necessarily identify the closest match and is sublinear. A key issue in addressing whether a search is an "approximate nearest neighbor search" is whether the search is designed to and

will necessarily identify an exact match or the closest match, or whether the search could identify search results that do not include an exact or the closest match. If a sublinear search can return a "close match that is not necessarily the closest match," it is an "approximate nearest neighbor search." But if a search cannot return a "close match that is not necessarily the closest match" (because it is designed to only find the closest match), then it is not an "approximate nearest neighbor search," irrespective of how the search is performed.

148. The input fault tolerance and skipped sections search features describe how a peak note search may be performed. Neither enables a peak note search to return a result other than the closest match. While the Petition identifies these two search features—the way the search is conducted—the Petition does not address the output of the searches much less identify a search that does not necessarily identify the closest match. As demonstrated above, the output from any disclosed Iwamura search always identifies the closest match and therefore is not an "approximate nearest neighbor search"—a search "identifying a close match that is not necessarily the closest match." Iwamura therefore does not disclose an approximate nearest neighbor search. I will specifically address each of the two search features identified by Petitioner is addressed in turn.

*input fault tolerance*

149. Iwamura discloses that its peak note search can include an "input fault tolerance." Iwamura, 9:20-24. Input fault tolerance allows a user to identify the closest match, even when the melody entered by a user has some errors. Iwamura, 9:33-39 (input fault tolerance enables "a correct search . . . notwithstanding inaccurate input from the user."). Using the fault tolerance feature, the peak note search first performs a search based on a tolerance of no errors, then a tolerance of one error, then a tolerance two errors, *etc*. The search will continue to search based on additional errors only if the search has not identified a match.

150. Accordingly, using the fault tolerance feature, the Iwamura search always produces an exact match or the closest match—it does not produce a result that is not necessarily the closest match. *See e.g.*, Iwamura, 11:43-45 ("The invented input fault tolerance function allows the user to obtain an exact result even when an entered melody has some errors."). Because the record identified using the fault tolerance search is necessarily the closest match, it is not a search that returns a "close match that is not necessarily the closest match," and, as a result, the feature does not disclose the claimed "approximate nearest neighbor search."

*<u>skipped sections</u>*

151. Iwamura also teaches that the disclosed search has "flexibility on search area" within a record in the reference database. Iwamura, 9:35. For

example, a user can identify the "important" portions of a melody, thereby enabling the search to skip the remaining "unimportant" portions. *See* Iwamura, 9:45-50 ("In a long music selection, there are some important portions that are indispensable to identify the melody. These portions are well recognized and remembered by the user. The user identifies such important portions as a keyword (key-melody). The other unimportant portions [in a long reference melody] can often be ignored."). The skipped sections feature is a pre-processing component, and what remains to be searched can be viewed as the "extracted" features over which an exhaustive search is performed until a match is found.

152. Iwamura does not disclose that flexibility on search area enables the Iwamura search to return a result other than the closest match. *See* Iwamura, 9:35-55. Because the record identified using the skipped portion search feature is still necessarily the closest match, it is not a search that returns a "close match that is not necessarily the closest match" and the feature does not disclose the claimed "approximate nearest neighbor search."

153. Moreover, as I explained above, an "approximate nearest neighbor search" is a sub-linear search, and each of the passages cited by Petitioner does not disclose a sub-linear search.

*Board's concerns*:

154. I now address the Board's specific concerns (identified in its Decision) with respect to whether Iwamura discloses the claimed "approximate nearest neighbor search." In instituting Ground 1, the Board preliminary found that Iwamura disclosed the "approximate nearest neighbor search" because the "approximate nearest neighbor search" "does not require that all of the records in the library are not used":

> deficiency with respect to the instant claims. With respect to "approximate nearest neighbor search," Patent Owner argues that the input fault tolerance capability of Iwamura cannot teach the same because it does not state or imply "that all records in the music library are not used in the comparison as required in an 'approximate nearest neighbor search.'" *Id*. at 19–20. Our construction of "approximate nearest neighbor search" to be "identifying a close match that is not necessarily the closest match" does not require that all of the records in the library are not used, so we also do not find this to be a deficiency of the ground.

Decision ('237) at 12. It is my opinion that the Board's preliminary analysis is flawed at multiple levels.

155. First, it is my understanding that the Board's preliminary analysis has the relevant burden backwards—it is not the Patent Owner's burden to demonstrate that the referenced "fault tolerance capability of Iwamura" does not disclose an "approximate nearest neighbor search." Rather it was the Petitioner's burden to demonstrate that Iwamura (and the "fault tolerance capability") discloses an

"approximate nearest neighbor search." As I demonstrated above, Petitioner did not satisfy this burden.

156. Second, as I demonstrated above, one skilled in the art would understand that there is no evidence that the referenced "fault tolerance capacity of Iwamura" teaches a search that identifies a close match that is not necessarily (*i.e.*, not guaranteed to be) the closest match rather than search that is guaranteed to identify the closest match. As I demonstrated above, the evidence confirms the opposite—that Iwamura finds "the closest melody from the database." Pet. ('237) at 8 (*quoting* Iwamura, 9:24-25).

157. Third, as I demonstrated above, an "approximate nearest neighbor search" is a sublinear search, and there is no evidence that the referenced "fault tolerance capability of Iwamura" teaches a sublinear search as the phrase is used in the context of the '237 Patent.

**3. nonexhaustive search (claim element 25(b)).**

158. As I explained above, a "nonexhaustive search" is "a search that locates a match without a comparison of all possible matches." Decision ('237) at 7.

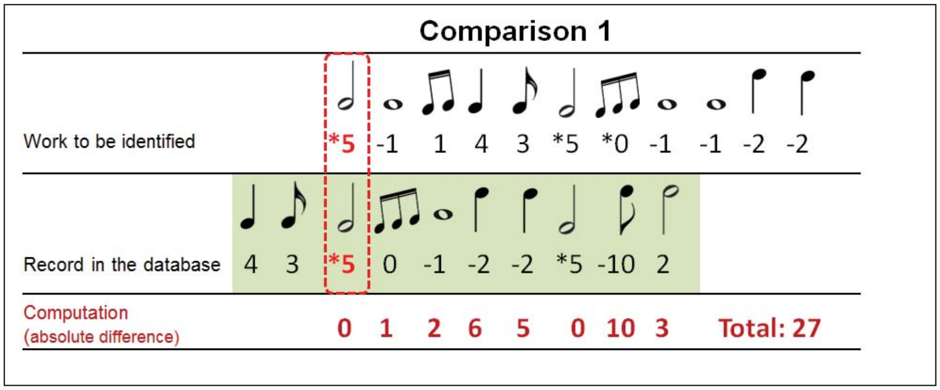
159. One skilled in the art would understand that Iwamura does not disclose a non-exhaustive search as the phrase is used in the context of the '237

Patent. As I described above, 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, 12:1-2. Instead of comparing the work to be identified with a record in the database by (a) preforming 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, Iwamura teaches that the shifting can be done peak-note-to-peak-note, thereby reducing the number of comparisons made between the work and a specific record, 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."

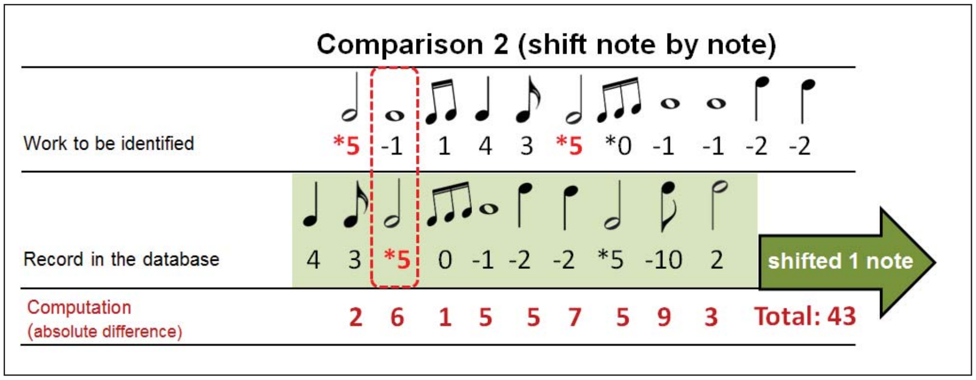
Iwamura, 9:9-11; *see* Iwamura, 5:9-13 ("The peaks in all the melodies stored in the databases are marked in advance. For melody matching, the entered melody is time-shifted . . . so that its peak matches each peak in the reference melody.").

160. This peak note search process can be illustrated using the example notes from Iwamura (Iwamura, 7:11-45). The following illustrates a first comparison between the notes from the work to be identified and the notes in a single record in the database:



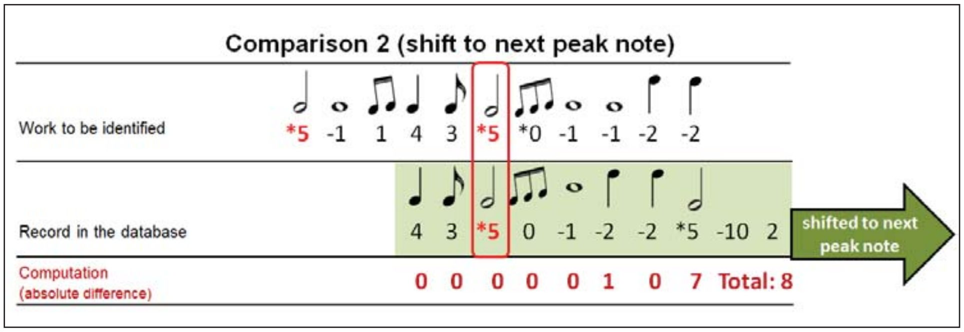

The top row represents the notes in the work to be identified; the middle row (highlighted in green) represents the notes of the record in the database being searched; and the bottom row (text in red) represents the absolute difference between the compared notes. The "peak notes" in the work to be identified and the record being searched are identified by "*". In this first comparison, the first peak note from the work to be identified (*5) and the record (*5) are aligned (as illustrated by the dashed red outline). Note that the computation (the absolute difference between the work to be identified and the record) results in a total value of 27 (0+1+2+6+5+0+10+3).

161. In a second comparison between the work to be identified and this same record in the reference database, the record in the database is shifted to the right by a single note (this is the "note by note" approach referenced in Iwamura (Iwamura, 9:9-11)):




The peak notes are not aligned in this comparison (as illustrated by the dashed red outline). The computation (the absolute difference between the work to be identified and the record) results in a total value of 43 (2+6+1+5+5+7+5+9+3).

162. An alternative to the second comparison presented above is to use the peak note approach taught in Iwamura. Using this peak note approach, the second comparison between the notes of the work to be identified and the notes in the record in the database is not just shifted one note to the right but is shifted to the right to align the next peek note (*i.e.*, five notes to the right), thereby skipping what would have been four intermediate comparisons using the alternative note by note approach:



As a result, the peak note approach taught in Iwamura avoided four unnecessary comparisons between the work to be identified and this reference work, making this peak note search more efficient. Note that the computation now results in a total absolute difference of 8. The number of comparisons that are avoided is 4 * (length of the query) as computing the individual distances between the notes requires a comparison.

163. 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, 7:53-55. Because the peak note search algorithm disclosed in Iwamura does not reduce the number of records to be searched or even the notes in each record to be searched but rather speeds up the individual comparison of the work to be identified to each record (by shifting the comparisons by peak notes rather than note by note), the disclosed algorithm searches all records in the library and is therefore an exhaustive search rather than the claimed "non-exhaustive" search. This approach does not reduce the number of records being searched *e.g.*, by discarding clusters of potential matches, like the sub-linear searches addressed in the IPR Patents. *See e.g.*, '237, 8:64-9:7 ("Other forms of matching include those based on <u>clustering, kd-trees, vantage point trees and excluded middle vantage point forests</u> are possible and will be discussed in more detail later. . . . Thus, for example, a <u>sub-linear search time</u> can be achieved.")

While the individual comparisons of a work and a record in the library can be more efficient using the peak note approach disclosed in Iwamura ("search speed can be increased"), in doing so each record in the library is searched as part of the disclosed algorithm and "[t]he reference melody that gives the least difference is returned as a search result." Iwamura, 7:53-55.

164. Accordingly, one skilled in the art would understand that Iwamura teaches an exhaustive search rather than the claimed "non-exhaustive" search, because it searches all records in the database using the peak note approach.

165. I note that Petitioner's Declarant, Dr. Moulin, 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):

> Q You would agree that in Iwamura, the
> search that's identified there does make a
> comparison to each of the possible musical works
> that could be returned as a match?
> MR. ELACQUA: Objection.
> THE WITNESS: To each of the musical works,
> yes.

Moulin Depo. 223:2-8.

> Q We've looked at each of the possible
> matches. We go through each record; right? Yes?
> A Yes.

Moulin Depo. 247:18-20.

Q Well, if it -- it's going to do a comparison to each reference work; right?
A That's correct.

Moulin Depo. 271:19-21.

Q Now, in the type of search that's identified and described in Iwamura, what it does is it does a comparison of the unknown melody to each of the melody patterns that are in the melody database; right?
A Yes.

Moulin Depo. 207:18-23. As a result, consistent with my understanding and the understanding of one skilled in the art, Petitioner's Declarant confirmed that, based on the proper construction of a non-exhaustive search (adopted by the Board), Iwamura does not disclose a non-exhaustive search:

Q By "nonexhaustive search," we mean a search that does a comparison but doesn't look at each of the melodies in our reference database. It's going to skip over some of them.
By "exhaustive search," we're going to go to each one of them. We might not use all the data for that melody, but we're going to do some comparison of the data to each of the melodies.
Do you understand?
A Yes.
Q With that definition, does Iwamura disclose a -- an exhaustive search or a nonexhaustive search?
A Well, with your, again, incorrect definition, that would be an exhaustive search. But, again, I disagree with your definition.

Moulin Depo. 233:24-234:14.

I want you to assume that by "exhaustive search," we mean a search that does a comparison to -- between an unknown work to each of the musical works in a database, but it doesn't require a comparison to all the data for each of the musical works.
And by a "nonexhaustive search," we mean one that does a comparison but doesn't look at each of the musical works. It just looks at some of them.
Would you agree that Iwamura -- with that definition -- teaches an exhaustive search?
A Under your incorrect premise, it would teach an exhaustive search.
Q By "incorrect premise," you mean the incorrect definition?
A Yes.

Moulin Depo. 225:16-226:7.

Q Before that search is run, each of those works is a possible match; right?
A Yes.
Q Would you agree that the Iwamura search, when it's run, it does a comparison of the unknown work to each of those possible matches?
A To each of those possible music works, yes. An approximate comparison, just to be clear.
Q And by "approximate," you mean that it doesn't necessarily look at every bit of data in every musical work?
A It does not necessarily -- exactly -- use all the data, and then it uses only approximations to the matching criterion.
Q But it does examine each of the possible musical works -- or each of the musical works that could be returned as a possible match?
A Yes.

Moulin Depo. 217:1-18.

166. I note that Petitioner's Declarant also confirmed that (as illustrated in the examples presented above) "all the notes" from each record in the database are compared. As a result, the searches disclosed in Iwamura would not be non-exhaustive even based on Petitioner's construction that includes the improper "and all data within all possible matches" clause (Pet. ('237) at 6):

Q I understand. But if we've got in our database a -- it's more than one peak. We've got a set of notes -- it doesn't say evaluate just one peak, does it?
A No.
Q We're evaluating all the notes in that phrase; right?
A Right. Right.

Moulin Depo. 280:6-13.

Q It says, "In this manner, the entered melody is shifted to each peak in each reference melody and compared."
Do you see that?
A Yes.
Q Does this indicate to you that Iwamura is teaching a peak search method in which it's going to compare the unknown melody with each peak in the -- each reference melody?
A In this case -- I'm just reading the context. Okay?
So all the notes are used. Okay. We are back to this same numerical example. So all the notes are used in this example, and therefore, he evaluates the -- you know, the least absolute error criterion.

Moulin Depo. 277:6-21.

167. The Petition, Declaration, and corresponding charts fail to demonstrate that Iwamura discloses a "nonexhaustive search." Petitioner and its

Declarant identify three features of the Iwamura search as teaching non-exhaustive searching:

(a) <u>peak notes</u>: a search that shifts the comparison of the notes in the work to be identified with the notes in a records by peak notes rather than note-by-note;

(b) <u>limit function</u>: comparing the work to be identified with a specific record in the database can be stopped and shifted to the next peak notes when the computation of the total absolute difference between the notes in the work to be identified and the specific record exceeds a certain limit;

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

Pet. ('237) at 9-10. Petitioner identifies these three features from Iwamura (labeled ❶, ❷, and ❸ below) as disclosing the non-exhaustive search in its Petition, Declaration, and corresponding charts, addressing either all three features or two of the features:

168. <u>Petition</u>:

> Claim 25 of the '237 patent further requires that the search is "nonexhaustive." Ex. 1001 at Claim 25. 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. In another example of non-exhaustive search, Iwamura teaches decreasing search time by stopping the search when computations "exceed[] a certain limit." Ex. 1012 at 7:56-57. In yet another example of non-exhaustive search, Iwamura discloses skipping "portions that should not be searched" (*id.* at 12:6-7), such as "repeated patterns" (*id.* at 9:36-44) and "unimportant portion[s]" of the melody (*id.* at 9:44-45).

Pet. ('237) 9-10.

169. Petition chart:

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a nonexhaustive search of reference extracted features of reference media works to identify a near neighbor; and | Petitioner incorporates the above discussion of Iwamura regarding Claim 9b. Iwamura further discloses non-exhaustive search algorithms using "peak notes" (6:31-7:55), which "are approximately 20% of the total number of notes in a typical melody," meaning "search speed using peak notes is 20% of a brute force search" (9:9-10). The search is further non-exhaustive because it can be accelerated by stopping the search when computations "exceed[] a certain limit." 7:56-57. |

Pet. ('237) 15.

170. Declaration:

> 71. 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." *See id.* 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.
>
> 73. 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.
>
> 74. 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 9:44-5) constitutes another example of non-exhaustive searching.

Moulin ('237) Decl. ¶¶71, 73-74.[16]

171. Declaration Chart:

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a nonexhaustive search of reference extracted features of reference media works to identify a near neighbor; and | I incorporate my above discussion of Iwamura regarding Claim 9b. Iwamura further discloses using non-exhaustive search algorithms using "peak notes" (6:31-7:55), which "are approximately 20% of the total number of notes in a typical melody," meaning "search speed using peak notes is 20% of a brute force search" (9:9-10). The search is further non-exhaustive because it can be accelerated by stopping the search when computations "exceed[] a certain limit." 7:56-57. |

16 Paragraph 72 of Dr. Moulin's Declaration addresses the "sublinear" rather than the "non-exhaustive" element.

Moulin Decl. (‘237) ¶75.

172. One skilled it the art would understand that none of these three Iwamura search features disclose the claimed “non-exhaustive search.” Each feature accelerates search speed within a single comparison of a work to be identified with a record in the reference database. No feature, however, enables the disclosed search to locate a match without comparing the work to be identified with each record in the reference database. I address each feature in turn.[17]

173. *peak notes*: A person of ordinary skill in the art would understand that the Iwamura “peak note” approach does not disclose a search that can locate a match without a comparison of all possible matches. As I explained above, a feature of the Iwamura search is that the search speed can be increased if the peaks of a melody input by a user are matched to the peaks of each reference melody, *i.e.*, each record in the reference database and the comparison between the work

[17] I observed that Petitioner’s Declarant also confirmed that another search feature disclosed in Iwamura—fault tolerance (that was not identified by Petitioner as support for the “non-exhaustive” search element)—also “does a comparison of the unknown work to each of the melodies in our reference database …it compares with every musical work, yes, in the database” and is therefore an exhaustive rather than non-exhaustive search. Moulin Depo. 268:15-20.

and the record is shifted by peak notes rather than note by note. *See* Iwamura 5:9-13 ("The peaks in all the melodies stored in the databases are marked in advance. For melody matching, the entered melody is time-shifted . . . so that its peak matches each peak in the reference melody.").

174. Peak note searching accelerates a search within a single comparison of the work to be identified with an individual record because, when comparing the notes of the work with the notes of the record, it shifts the notes to be compared by peak notes rather than note by note:

> "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, 9:8-11.

175. While this search technique may be efficient, the peak note searching disclosed in Iwamura still requires exhaustively searching every reference melody. Iwamura, 9:11-13 (discussing a faster comparison of "each reference melody" with respect to peak note searching); *see also* Iwamura, 7:52-54 (noting that in the search process, "the entered melody is shifted to each reference melody and compared"). As Petitioner's Declarant repeatedly confirmed (consistent with my understanding) "you search through every musical work in the database" for "all

the Iwamura searches" (including the "peak note" approach) (Moulin Depo. 269:19-270:2):

> Q Would you agree that in Iwamura's peak search note, that it does search across each melody in the melody database?
>
> A Yes, it does. It's going to compare it with every song in the database, typically.

Moulin Depo. 213:23-214:2; 223:2-8; 247:18-20; 271:19-21.

176. Under the proper construction of "non-exhaustive," the "peak note" approach and the corresponding passages from Iwamura cited in the Petition and Declaration do not disclose a non-exhaustive search because they do not state or suggest that all references in the music library are not compared. Rather, all reference melodies are compared and "[t]he reference melody that gives the least difference is returned as a search result." Iwamura, 7:52-55. Accordingly, a search using "peak notes" is not a non-exhaustive search.

177. Moreover, even applying the "all data" clause in Petitioner's improper construction—a non-exhaustive "search … locates a match without conducting a brute force comparison of … all data within all possible matches"—the peak note search disclosed in Iwamura is still an exhaustive (rather than non-exhaustive) search because it compares "all data within all possible matches." When comparing a work to be identified with each potential match, the peaks of the song

to be identified are lined up with the peaks of the reference work to expedite the comparison: "In this manner, the entered melody is shifted to each peak in each reference melody and compared." Iwamura, 7:52-55. But in doing so, this does not mean that only the peaks from the work to be identified are compared to the peaks of the reference work. Rather, once the peaks are lined up, both the peaks and valleys (all data) are compared in the computation. Dr. Moulin, at his deposition, agreed with this understanding of the peak note search:

```
        Q   It says, "In this manner, the entered
melody is shifted to each peak in each reference
melody and compared."
            Do you see that?
        A   Yes.
        Q   Does this indicate to you that Iwamura is
teaching a peak search method in which it's going to
compare the unknown melody with each peak in the --
each reference melody?
        A   In this case -- I'm just reading the
context.  Okay?
            So all the notes are used.  Okay.  We are
back to this same numerical example.  So all the
notes are used in this example, and therefore, he
evaluates the -- you know, the least absolute error
criterion.
```

Moulin Depo. 277:6-21.

178. While the Petition (Pet. ('237) at 5) quotes a passage from Iwamura that suggests Iwamura avoids a "brute force" search, one skilled in the art would

understand that the "brute force" being avoided (and what makes the algorithm efficient) is that peaks are not compared to valleys and valleys are not compared with peaks. Instead, by lining up the peaks when comparing the data, peaks are compared with peaks and valleys are compared with valleys. Therefore, when Iwamura states that its approach is 20% more efficient than a brute force search, one skilled in the art would understand that this does not mean that the peak note approach disclosed in Iwamura does not consider "all possible matches" or even "all data in all possible matches." Rather, it means that by lining up the peaks when doing the comparison, it will save time over comparing the music to be identified with the referenced song without first lining up the peaks; shifting the comparisons by peak notes is more efficient than simply shifting the comparisons "note by note." Iwamura, 9:8-11.

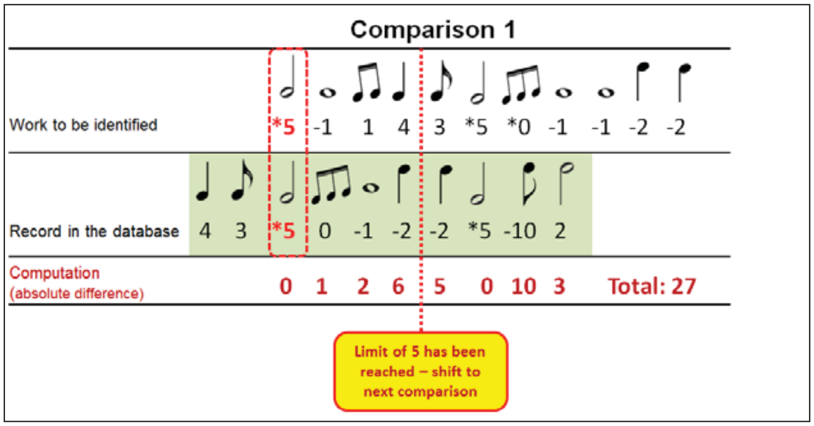
179. *limit function*: One skilled in the art would understand that the limit function approach addressed in Iwamura does not disclose non-exhaustive search under either the proper construction or under the Petitioner's flawed construction. Under the proper construction, a non-exhaustive search locates a match without comparing the work to be identified with all possible records in the reference database. The Iwamura limit function is not a search that locates a match without comparing the work to be identified with all possible matches. The Iwamura limit function accelerates the process of comparing the work to be identified to a single

record in the reference database. The limit function describes the ability of a user to input a "limit" whereby a computation based on comparing the notes of the work to be identified with the notes of an individual record for a particular peak will be stopped and shifted to the next peak for that record when the total absolute difference between the compared notes exceeds a certain value. Iwamura, 7:56-58.[18] Nothing in Iwamura talks about absolute distance calculated for a record (*i.e.*, across all peaks in the record), only for each peak.

180. For example (using the examples provide in Iwamura, 7:11-45), assume a user inputs a limit where the computation comparing the notes of the work to be identified and a single record in the database would be stopped when the total absolute difference in the computation exceeds 5:

---

18 Computation refers to the process of comparing the absolute difference between the integer values assigned to the notes in the work to be identified (the melody input by the user to be identified) and a single melody record in the reference database for a specific peak comparison.




This comparison would be stopped before all notes have been compared for this specific alignment because comparing the first four notes results in a computation of an absolute difference that exceeds the limit of 5:  0 + 1 + 2 + 6 exceeds the set limit of 5.

181. Once a peak range search is stopped by the limit function (*i.e.*, the total absolute difference exceeds a certain limit so that the computation is stopped), the search shifts to the next peak range comparison within the same record, and continues the search process until each peak in each record is compared against the melody input by the user.  A search that uses the limit function disclosed in Iwamura will still compare every record in the reference database:  "In this manner, the entered melody is shifted to each peak in each reference melody and compared.  The reference melody that gives the least different is returned as a search result."  Iwamura, 6:31-7:55.  My understanding of how the limit function of Iwamura works was confirmed by Petitioner's Declarant:

Q Now, if we -- and then, as you understand
it, what the search algorithm would do, it would
then shift this peak over to the next peak and start
another calculation; is that right?
A Yes. Yes.

Moulin Depo. 241:24-242:2.

182. Petitioner's Declarant confirmed that "you search through every musical work in the database" for "all the Iwamura searches" (including the limit function approach). Moulin Depo. 269:19-270:2.

Q -- when we throw in this limit on the
computation, we are going to still do a comparison
to each of the works that are in our database;
right?
A We do a comparison of some data within
each musical work, yes.
Q You don't read this as saying that if
we're doing a computation, and for one of them it
exceeds a certain limit, then we stop the search
altogether? It doesn't say that, does it?
A No. We move to the next one.
Q By "the next one," you mean the next
computation?
A The next possible match. Like after the
first line in your example, we move to the second
one.

Moulin Depo. 243:17-244:7.

Q If we use this parameter in doing our
search in Iwamura, is it the case that after we
determine that one calculation should be stopped,
then that we stop the search altogether, or do we
keep calculating --
A No. We move to the next one. So like
here, if the parameter value is 20, once we reach
20, we abandon this and move on to the next one.
Q Then after we complete the peak search
analysis for a given work, then we go to the next
work; is that right?
A Yes.

Moulin Depo. 242:19-243:5.

183. One skilled in the art would understand that the limit function search disclosed in Iwamura is therefore exhaustive.

184. Moreover, even using the "all data" clause from Petitioner's improper construction, one skilled in the art would understand that the limit function algorithm disclosed in Iwamura is still exhaustive rather than non-exhaustive because it compares "all data within all possible matches." While the search comparing a particular peak pattern of a work against a record can be stopped if the difference exceeds a certain limit, this does not mean that the comparison of the work with the record stops. Rather, as I described above, this means that the data in the work will be shifted against the record to match up with the next peak and the comparison of all the data will continue. Nothing in Iwamura expressly states

that all data will not be searched and a search that does not compare all data is also not inherent (*i.e.*, necessarily present).

185. *unsearched portions*: One skilled in the art would understand that this unsearched portion approach disclosed in Iwamura does not disclose a non-exhaustive search. If a search compares the work to be identified to each reference in a database, it is not the claimed non-exhaustive search. Even if certain portions of a reference are skipped, the unsearched portions approach of Iwamura still compares the work to be identified with all potential matches. Consistent with my understanding, Petitioner's Declarant confirmed that all musical records in the reference database are searched under all Iwamura searches (including the unsearched portions approach):

> Q Is it true that for all the Iwamura searches, all the variations that Iwamura teaches, it always teaches doing a comparison to each of the musical works that's a possible match in our database?
>
> A I don't think it says it explicitly every time. It's -- it's often implicit. It's understood that you search through every musical work in the database.

Moulin Depo. 269:19-270:2.

186. Petitioner's Declarant specifically confirmed, consistent with my understanding, that all potential matches in the database are searched using Imamura's unsearched portions approach:

> Q All right. So let's assume we've got our database up. We've identified the unimportant sections, and we're not going to assess those when we're doing our Iwamura search.
>
> Does that make sense?
>
> A Yes.
>
> Q In that case, failing to test a melody against an unimportant part will not result in us ignoring a match. Would you agree?
>
> A That is correct, assuming it's truly an unimportant part, yes.

Moulin Depo. 317:2-12.

> Q Now, one of the features of Iwamura is an alternative in which the database can be set up where you're going to not -- you're going to skip portions of the reference melodies. You're only going to use what he identifies as the important parts; right?
>
> A Yes.
>
> Q In that -- when that version of Iwamura is used, is it the case that the Iwamura search will do a comparison to each of the musical works that are in the database?

Moulin Depo. 267:13-24.

187. Iwamura does not expressly state (nor is it inherent, *i.e.*, necessarily present) that flexibility on search area enables the disclosed search to entirely skip a record in the reference database. Each and every record in the reference database will be searched; therefore, the search is an exhaustive search rather than the claimed non-exhaustive search. Moreover, when a repeated pattern (*e.g.*, "second measure") is skipped, it is a "reasonable engineering assumption" that the search has "already tested" the repeated pattern and, as a result, all data is considered in the search. Moulin Depo. 279:7-14. Moreover, Iwamura states that each repeated portion can be pre-processed and is marked as such in the database. Iwamura, 9:39-42. Accordingly, the unsearched portion process constitutes extracting the features of the melodies to be compared and the resulting search searches all pre-processed data.

188. <u>*Board's concerns*</u>: I now address the Board's specific concerns (identified in its Decision) with respect to whether Iwamura discloses the claimed nonexhaustive search. In instituting Ground 1 of the '237 Petition, the Board determined that one feature of Iwamura identified by Petitioner—the "computational limits" feature—discloses a non-exhaustive search because if the computation limit (comparing the notes in the work to be identified with a single record in the database) is reached, the entire search is stopped, independent of how many records in the database have actually been searched:

> Patent Owner also argues that Iwamura's computational limit does not create a nonexhaustive search because "it does not state or suggest that all records in the music library are not use[d] in the comparison." Prelim. Resp. 18. 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 "nonexhaustive 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 would ultimately provide a match because of the input fault tolerance process, discussed above. *See* Ex. 1012, 7:56–57, 9:20–34.

Decision ('237) at 11-12. In making this preliminary finding, it appears that 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 for a specific alignment of peak notes and then shifting the peaks to perform another peak comparison with that record, with

(b) stopping the entire search process altogether.

In my opinion, there are at least two reasons why the Board's preliminary interpretation of Iwamura is not correct.

189. Reason 1: Iwamura does not state (or even suggest or imply) 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 entire search stops. Neither the Petition, Petitioner's Declarant, nor the Board points to such a statement in Iwamura, because one skilled in the art would understand that there is none. Rather Iwamura states that to accelerate comparing the peaks of the work to be identified with a single record in the database, the "computation of the total absolute difference" between the melody and a specific reference work based on that search can be stopped and shifted to the next comparison:

> To accelerate the search, computation of the total absolute difference can be stopped when it exceeds a certain limit.

Iwamura, 7:56-57.

190. The individual computation based on that particular alignment between the peak notes of the work to be identified and the record "can be stopped" when that individual computation exceeds a certain limit. The search process itself is not stopped but rather accelerated: "[t]o accelerate the search." "In this manner, the entered melody is shifted to each peak in each reference melody and compared. The reference melody that gives the least difference is returned as a search result. Iwamura, 6:31-7:55. The specific computation is stopped, not the search: "it would then shift this peak over to the next peak and start another calculation." Moulin Depo. 240:24-242:2.

191. Again, I note that, consistent with my understanding, Petitioner's Declarant confirmed that under the computation limits approach disclosed in Iwamura (as well as all other approaches in Iwamura), all potential matches are searched:

> Q Is it true that for all the Iwamura searches, all the variations that Iwamura teaches, it always teaches doing a comparison to each of the musical works that's a possible match in our database?
>
> A I don't think it says it explicitly every time. It's -- it's often implicit. It's understood that you search through every musical work in the database.

Moulin Depo. 269:19-270:2.

Q -- when we throw in this limit on the computation, we are going to still do a comparison to each of the works that are in our database; right?
A We do a comparison of some data within each musical work, yes.
Q You don't read this as saying that if we're doing a computation, and for one of them it exceeds a certain limit, then we stop the search altogether? It doesn't say that, does it?
A No. We move to the next one.
Q By "the next one," you mean the next computation?
A The next possible match. Like after the first line in your example, we move to the second one.

Moulin Depo. 243:17-244:7.

Q If we use this parameter in doing our search in Iwamura, is it the case that after we determine that one calculation should be stopped, then that we stop the search altogether, or do we keep calculating --
A No. We move to the next one. So like here, if the parameter value is 20, once we reach 20, we abandon this and move on to the next one.
Q Then after we complete the peak search analysis for a given work, then we go to the next work; is that right?
A Yes.

Moulin Depo. 242:19-243:5.

192. *Reason 2*: One skilled in the art would understand that the alternative (which is not disclosed in Iwamura)—that the entire search process stops when one peak search comparison between the work to be identified and one record in the database reaches a certain limit—would make the search process inoperable. The purpose of Iwamura is to find a match. Stopping the search when an individual computation exceeds a certain limit would prevent the search from finding a match. For example, assume that:

- there are 10 records in the dataset to be search;
- the computation based on the first peek note alignment between the work to be identified and the first record in the database exceeds the set limit.

Stopping the search at that point—after comparing the work to be identified with just the first alignment of the first record—would identify no match even if records 4, 7, and 8 were close matches and record 9 was an exact match. The system would be inoperable and would fail to identify matches if the search is stopped completely when a computational limit is reached rather than, as disclosed in Iwamura, the search moves on to (a) the next alignment of peak notes between the work to be identified and that same record in the database, or (b) the next potential record in the database to identify a match. Stopping the search when a given computation exceeds a certain limit will speed up comparing the work to be identified with a given record in the database but it does not stop the search process.

193. I note that the Board also noted that if Iwamura disclosed a search that is not a nonexhaustive search, this "does not end the inquiry"—Iwamura could still teach a nonexhaustive search as long as, in addition to disclosing other searches, Iwamura actually disclosed the claimed nonexhaustive search:

> We note that all of the independent claims of the '237 Patent utilize "comprising" language, such that those claimed methods and apparatuses do not exclude additional, unrecited elements or method steps. *See Mars Inc. v. H.J. Heinz Co.,* 377 F.3d 1369, 1376 (Fed. Cir. 2004). Thus, the scope of independent claim 25 can include an exhaustive search, as long as it performs a nonexhaustive search as well. Thus, even if Patent Owner is correct and a particular search in Iwamura is exhaustive, that does not end the inquiry. In addition, we note that no claim in the '237 Patent requires the

Decision ('237) at 11. As I demonstrated above, one of ordinary skill in the art would understand that Iwamura does not disclose any nonexhaustive searches.

#### 4. identify a neighbor / near neighbor (claims elements 1(b), 5(b), and 25(b)).

194. In instituting Ground 1, the Board did not specifically address whether Iwamura disclosed the neighbor or near neighbor properties of the claimed search. Decision ('237) at 11-12. As I demonstrated below, one skilled in the art would understand that Iwamura does not disclose such properties.

195. As I explained above, identifying “a neighbor” or “near neighbor” means identifying “a close, but not necessarily exact or closest, match.” Decision (‘237) at 8.

196. Iwamura does not disclose a search to identify a neighbor or near neighbor because, as I explained above, the disclosed search always identifies an exact or the closest match. Iwamura confirms that the disclosed search engine will find the “closest” match—the melody that gives the least difference. Iwamura, 9:54-55. Petitioner’s Declarant also confirmed that Iwamura will either produce an “exact match” if it finds one, or the “best match it finds using that approximate criterion.” Moulin Depo. 271:22-272:12.

197. One skilled in the art would understand that the system in Iwamura will always find the closest match, even if unimportant peaks are skipped or repeated patterns are avoided. At his deposition, Dr. Moulin agreed to my understanding:

- “[W]’re still going to be identifying the closest match” even when “the unimportant peaks are skipped…. Dropping an unimportant part is not going to affect the ability to find the best match.” Moulin Depo. 317:14-22.

- "If we implement that feature of Iwamura… skipping a repeated pattern…. It will not affect the ability to find the best match." Moulin Depo. 318:11-18.

Because the search algorithms disclosed in Iwamura necessarily return the closest match, they do not identify a match that is not necessarily the closest match, as the neighbor and near neighbor claim elements require. Iwamura does not disclose identifying a neighbor or near neighbor because the disclosed search always identifies an exact or the closest match.

198. The Petition, Declaration, and corresponding charts fail to demonstrate that Iwamura discloses the claimed neighbor or near neighbor searches.

199. The Petition does not address the "neighbor" concepts in the text of the Petition. In its Charts, to establish the claimed search "to identify a neighbor" (elements 1(b) and 5(B.2)) and search "to identify a near neighbor" (element 25(b)), Petitioner asserts:

Claim 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Iwamura determines an identification of the media work using the extracted features by "find[ing] the closest melody from the database," which is a neighbor. 9:25-38, 12:1-2. Iwamura discloses searching using the "Boyer-Moore algorithm" (9:63-64, 10:1-3), which is sublinear (Ex. 1017 at 1). Ex. 1004 at ¶ 72. |

Pet. (‘237) at 10-11.

Claim 5(b.2):

| | |
|---|---|
| 2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and | Petitioner incorporates the above discussion of Iwamura regarding Claim 1b. |

Pet. (‘237) at 12.

Claim 25(b): Petitioner incorporates its discussions regarding element 9(b) (the remaining discussion addresses the non-exhaustive component of the claim element):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a nonexhaustive search of reference extracted features of reference media works to identify a near neighbor; and | Petitioner incorporates the above discussion of Iwamura regarding Claim 9b. Iwamura further discloses non-exhaustive search algorithms using "peak notes" (6:31-7:55), which "are approximately 20% of the total number of notes in a typical melody," meaning "search speed using peak notes is 20% of a brute force search" (9:9-10). The search is further non-exhaustive because it can be accelerated by stopping the search when computations "exceed[] a certain limit." 7:56-57. |

Pet. (‘237) at 15.[19]

For claim element 9(b), Petitioner asserts:

[19] The referenced claim element 9(b) does not include a search “to identify a near neighbor” but instead includes “an approximate nearest neighbor search.”

| | |
|---|---|
| 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 Iwamura regarding Claim 1b. Furthermore, Iwamura uses an approximate nearest neighbor "search engine [that] has . . . input fault tolerance capability" (10:17-18), and skips "portions that should not be searched" (12:6-7), such as "repeated patterns" (9:36-44), and "unimportant portion[s]" of the melody (9:44-45). |

Pet. ("237) at 12.

200. The Declaration is essentially the same.

Claim element 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Iwamura discloses the use of a "search engine" to determine an identification of the media work using the extracted features by "find[ing] the closest melody from the database," which is a neighbor. 9:25-38, 12:1-2. Iwamura discloses searching using the "Boyer-Moore algorithm" (9:63-64, 10:1-3), which is sublinear (Ex. 1017 at 1). |

Moulin Decl. ('277) ¶75.

Claim element 25(b) cross references claim element 9(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a nonexhaustive search of reference extracted features of reference media works to identify a near neighbor; and | I incorporate my above discussion of Iwamura regarding Claim 9b. Iwamura further discloses using non-exhaustive search algorithms using "peak notes" (6:31-7:55), which "are approximately 20% of the total number of notes in a typical melody," meaning "search speed using peak notes is 20% of a brute force search" (9:9-10). The search is further non-exhaustive because it can be accelerated by stopping the search when computations "exceed[] a certain limit." 7:56-57. |

Claim element 9(b):

| | |
|---|---|
| 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 Iwamura regarding Claim 1b. Furthermore, Iwamura discloses using an approximate nearest neighbor "search engine [that] has . . . input fault tolerance capability" (10:17-18), and skips "portions that should not be searched" (12:6-7), such as "repeated patterns" (9:36-44), and "unimportant portion[s]" of the melody (9:44-45). |

201. One skilled in the art would understand that these discussions and the cited passages from Iwamura do not demonstrate that Iwamura teaches a search that identifies a neighbor or near neighbor for the reasons that I set forth above.

202. First, the cited passage from element 1(b) does not disclose a search that identifies a neighbor or near neighbor. As I explained above, a search that identifies a neighbor or near neighbor is a search that identifies "a close, but not necessarily exact or closest, match." Decision ('237) at 8. The passage cited in the Petition and corresponding declaration confirms that the Iwamura searches find "the closest melody from the database." Pet. ('237) at 8 *(quoting* Iwamura, 9:24-35).

203. Second, the references to searches that have an "input fault tolerance" or skip "portions that should not be searched" (Pet. ('237) 13 quoting Iwamura 10:13-18, 12:6-7, 9:36-44, and 9:44-45) do not expressly or inherently disclose a search that does not necessarily identify the closest match. As I demonstrated above, the output from any disclosed Iwamura search always identifies the closest match and therefore is not a search that identifies a neighbor or near neighbor—"a

close, but not necessarily exact or closest, match." *See e.g.*, Iwamura, 11:43-45 ("The invented input fault tolerance function allows the user to obtain an exact result even when an entered melody has some errors.").

### 5. sublinear approximate nearest neighbor search (claim element 33(b)).

204. Claim 33 requires a search that is both (a) a sublinear, and (b) an approximate nearest neighbor search.

205. One skilled in the art would understand that Iwamura does not disclose a "sublinear approximate nearest neighbor search" for at least two independent reasons.

206. Reason 1: As I demonstrated above (with respect to claim elements 1(b) and 5(b.2)), Iwamura does not disclose a "sublinear" search.

207. Reason 2: Also as I demonstrated above (with respect to claim elements 9(b) and 13(b.2)), Iwamura does not disclose an "approximate nearest neighbor search."

208. The Petition, Declaration, and corresponding charts fail to demonstrate that Iwamura discloses the claimed "sublinear approximate nearest neighbor search." For claim 33, the Petition and corresponding Declaration do not address the "sublinear approximate nearest neighbor search" in their respective

texts but instead exclusively cross-reference their respective charts for Claims 1(b) and 9(b).

Petition:

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a sublinear approximate nearest neighbor search of reference extracted features of reference identified media works; and | Petitioner incorporates the above discussion of Iwamura regarding Claims 1b and 9b. |

Pet. (‘237) at 16.

Declaration:

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the media work extracted features to perform a sublinear approximate nearest neighbor search of reference extracted features of reference identified media works; and | I incorporate my above discussion of Iwamura regarding Claims 1b and 9b. |

Moulin Decl. (‘237) ¶75. As I demonstrated above, the cross-referenced “discussions” and citations to Iwamura fail to demonstrate that Iwamura discloses either a “sublinear” search or an “approximate nearest neighbor search.” Accordingly, the Petition fails to satisfy its burden for these two independent reasons.

209. Board’s concerns: I addressed the Board’s concerns with respect to the “sublinear” component above in Section VI(A)(1). I addressed the Board’s

concerns with respect to the "approximate nearest neighbor component" above in Section VI(A)(2).

### B. '237 Ground 2: The instituted claims of the '237 patent are not anticipated by Ghias.

210. The Board instituted Ground 2 based on the following: Claims 1–3, 5–7, 9–11, 13–15, and 21–24 as unpatentable under 35 U.S.C. § 102(b) as anticipated by Ghias. Decision ('237) at 21 (I underlined the independent claims). Ground 2 fails because Ghias does not disclose the following key elements from each instituted independent claim:

- sub-linear time search (claim elements 1(b) and 5(b.2)); and
- approximate nearest neighbor search (claim elements 9(b) and 13(b.2)).

I address each in turn below.

#### 1. sublinear time search (claim elements 1(b) and 5(b.2)).

211. Claims elements 1(b) and 5(b.2) require a "sub-linear time search."

212. As I explained above, one of ordinary skill in the art would understand that a "sub-linear time search" is "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched." Decision ('237) at 7.

213. Ghias does not disclose a "sub-linear time search" search but instead teaches a linear search in which the search time grows linearly in relationship to the size of the data set. The searches disclosed in Ghias compare the work (user input 23) with "all the songs" in the library (*i.e.*, what the Petition calls "all possible matches," Pet. 6):

> 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.[20] If an increase in a given variable increases the execution time of a given algorithm by an amount that is only a constant multiple of the amount by which that variable was increased, irrespective of the initial value of that variable, then that algorithm scales linearly with regard to that variable.[21] More specifically,

[20] To compare the "user input" with "all the songs," Ghias must compare the user input with every song in the data set. Ghias does not disclose a search algorithm that does not compare the work to be identified with every record in the data set.

[21] As I explained above, linearity describes "[t]he relationship existing between two quantities when a change in a second quantity is directly proportionate to a change in the first quantity." Ex. 2007 (Modern Dictionary of Electronics) at 425 (1999).

if f(n1, …, ni, …, nk) is a function that describes the execution time of an algorithm where variables n1, …, ni, …, nk are the sizes of the different types of data on which the algorithm operates, then if f(n1, …, ni+q, …, nk) = f(n1, …, ni, …, nk) + f(n1, …, q, …, nk), then that algorithm scales linearly with regards to variable ni. Because a constant increase in the size of the data set (*i.e.*, number of records in the reference data set) increases the execution time of the Ghias search algorithm by a constant amount that does not depend on the initial size of the data set, Ghias discloses a linear time search, not a sub-linear time search.

214. In addressing "the problem of approximate string matching," Ghias identifies "the running times of several algorithms:

> 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.

Ghias, 6:23-28. In each instance, the running time of the identified search is linear (not sub-linear) with respect to the size of the data set.

215. As clarified in this passage from Ghias:

- "m is the number of pitch differences in the query" corresponding to the length of the query of the work to be identified (highlighted in green); and

- "n is the size of the string (song)" (highlighted in orange); Moulin Depo. 88:13-15.[22]

- "k" refers to the number of mismatched characters permitted in the search results returned by the search: "The problem consists of finding all instances of a pattern string P = p1, p2, p3 . . . pm in a text string T = t1, t2, t3 . . . tn such that there are at most k mismatches (characters that are not the same) for each instance of P in T." Ghias, 6:37–41; Moulin Depo. 96:2-15.

- In the field of computer software, "O" indicates big O notation. Big O notation describes the relationship between an algorithm's execution time and other variables. In computer science, big O notation is used to describe how algorithms respond (*e.g.*, in their processing time or working space requirements) in the worst-case to changes in input size. Ex. 2009 (http://en.wikipedia.org/wiki/Big_O_notation.); Moulin Depo. 16:13-24 (in the field, there is "a common system of notation that's used … when we're taking about how the search time or execution time scales with respect to the size of the database – it's the so-called order of notation … sometimes referred to as the 'big O notation.'")

216. As I explained above, the disclosed searches may be sublinear with respect to "m … the number of pitch differences in the query." O((nlog(m))) is

[22] Referring to "n" as part of the dataset to be searched (rather than the query of the work to be identified) is standard in the field. Moulin Depo. 18:2-10.

sublinear with respect to "m" (corresponding to the length of the query) because the running time is a function of log(m)). *See* Moulin Depo. 102:9-13.[23]

217. The disclosed searches are never sublinear with respect to "n"—"the size of the string (song)" or the number of records in the data set ("N").[24] As I explained above, if a constant increase in a given variable increases the execution time of a given algorithm by a constant amount, then that algorithm scales linearly with regard to that variable. An incremental increase in the number of records in the data set, or even in the length of a given reference record ("n") in the data set, increases the execution time of every search disclosed by Ghias by a constant amount.

218. O(mn), O(kn), and O(nlog(m)) all describe algorithms whose execution times increase by a constant amount as the length of the record being searched is incrementally increased. The first to run times—O(mn) and O(kn)—are linear with respect to the size of the data set being searched. My

---

[23] "log" stands for taking the logarithm of the following variable; so log(m) means the logarithm of m.

[24] As I noted above, in the IPR Patents, consistent with the literature, the size of the dataset is referred to as "N" where "N" is the number of records in the dataset.

understanding, consistent with the understanding of one skilled in the art, is confirmed by Petitioner's Declarant, *see, e.g.*

> Q Let's assume we have a search that's
> execution time is O(mn), where M means the size of
> the query, N means the size of the database or
> dataset that we're searching over.
> What does that tell us?
> A Well, it means that the search time,
> f(m)n, grows at most linearly.
> Q Is it the case that this would be a
> sublinear search as it's used in the '237 patent?
> A No. It says, again, it's at most linear
> in terms of M times N.

Moulin Depo. 28:17-29:2.[25] The third run time—O(nlog(m))—may be sub-linear with respect to the number of pitch differences in the query "m" but is always linear with respect to "n," the size of the string (song) being searched, or the number of records in the dataset being searched. Again, my understanding, consistent with the understanding of one skilled in the art, is confirmed by Petitioner's Declarant *see, e.g.*:

---

[25] Petitioner's Declarant uses "data set" and "database" interchangeable in this context. Moulin Depo. 22:14-16.

> If we have a search that is described as set forth in Formula 6, where N refers to the size of the database, what does that tell us about whether it's linear or sublinear?
>
> A Same answer as previously. You have replaced f(m) with log(m). It's just a particular function, f(m). So that would tell me that search time is at most linear in terms of N.
>
> Q If you're presented with that, does that tell you that we have a sublinear search?
>
> A In terms of N, yes.
>
> Q In terms of --
>
> A Sorry. Let me rephrase this.
>
> It's at most linear.
>
> Q Does that tell us that we have a sublinear search?
>
> A It does not tell me that, because of the meaning of the order of notation. It says, at most, linearly.

Moulin Depo. 36:20-37:13.[26]

219. Accordingly, Ghias exclusively disclose searches that are linear—not sublinear—in relationship to the data set to be searched. My understanding is again confirmed by Petitioner's Declarant:

---

[26] These "running times" are the times it takes to run a query of length "m" against one record of the length "n" in a dataset including N records. The search time for running the same query against the full dataset would take on average N time longer, since each record in the dataset will need to be searched.

Q Does the information that's presented here suggest to you that these -- that these algorithms are sublinear with respect to the size of the dataset being searched?
A It does not say that. First, there's no database. Okay? Again, this refers to a single song.

Moulin Depo. 88:22-89:3.

Q Reading this, do you interpret this to suggest that these algorithms are sublinear with respect to the size of the dataset being searched?
A No. This, again, reads "of the order of." So it means at most linear.

Moulin Depo. 89:14-18.

Q My question, sir, is did you read this to indicate or suggest that these algorithms would -- could be used to perform a sublinear time search with respect to the size of the dataset being searched?
A No.

Moulin Depo. 90:16-21; 93:24-94:5; 98:20-25; 100:8-11; 142:5-10.

220. Petitioner's Declarant confirmed my understanding—that:

(a) any sub linearity referenced in Ghias is with respect to "m"—the number of pitch differences in the query, not "n" the size of the string (song) or the size of the data set (N);

(b) Ghias does not state or suggest that the size of the query is dependent on the size of the data set;

(c) any sub linearity with respect to the query "is not relevant" to the '237 patent, and

(d) as a result, Ghias does not disclose a search that is sublinear with respect to the size of the data set—the relevant sub-linearity inquiry for the '237 patent.

Moulin Depo. 152:20-154:2 (any sub-linearity with respect to the query "is not relevant.")

221. In reviewing Dr. Moulin's deposition, I observed that Petitioner's Declarant, Dr. Moulin, testified that:

(1) he clearly understood that sub-linear in the context of the '237 patent is based on the size of the data set searched, not the size of the query or the pitch differences in the query;

(2) Ghias does not identify a search that is sub-linear with respect to the data set; and

(3) when he wrote in his Declaration that "Ghias discloses searches … which … are sublinear," he did not intend the Board to interpret "sublinear" to be in the context of the '237 patent but rather in a different context irrelevant to the '237 patent.

222. As I noted above, Petitioner's Declarant understood that, consistent with my understanding, "sublinear" in the context of the '237 patent ("a concept that's common in [his] field" (Moulin Depo. 8:10-14)) is based on the size of the data set searched (N), not the size of the query or pattern to be matched ("m"):

> 53. I understand and agree with Petitioner's position that the term "sublinear search" means "a search whose execution time has a sublinear relationship to database size." For instance, a linear search of a 200-item database would take twice as long as a linear search of a 100-item database. By contrast, a sublinear search of a 200-item database would take less than twice as long as a sublinear search of a 100-item database, perhaps, for instance, 1.5 times as long.

Moulin Decl. ('237) ¶53.

Is it the case that for a search to be
sublinear as it's used in the '237 patent, it's not
enough for it to have execution time that is
sublinear in relationship to the size of the
pattern; it must also be sublinear in relationship
to the size of the database?
A When I read "sublinear" in, say, Claim 5
of the patent, as we just did, I understand
sublinear to mean in relation with the size of the
database. It does not say anything about in
relation with the size of the query.

Moulin Depo. 26:11-21.

Q Is it the case that in doing your analysis
to determine whether or not prior art anticipated,
you applied the definition that a sublinear search
or a sublinear time search was one whose execution
time has a sublinear relationship to the database
size?
A That is correct, yes.
Q Is that the correct definition that should
be applied, or are you applying the wrong
definition?
A In my opinion, this is the correct
definition to be used.

Moulin Depo. 24:1-12.

> Q Well, my question is if we're trying to determine whether or not a given search is sublinear with respect to the size of the database, that means that we're going to determine execution time compared to the growth of the database, N; right?
>
> A Yes.

Moulin Depo. 31:13-18.

> Q Well, as you understood the term "sublinear" as it's used in the patent, it's about the execution time as we increase the size of the database; is that right?
>
> A In the patent, yes.
>
> Q Yes.
>
> A Yes.

Moulin Depo. 103:16-22.

223. Petitioner's Declarant agreed with my understanding that the algorithms disclosed in Ghias do not disclose a search that is sublinear with respect to the size of the data set but instead has a sub-linear relationship to "m" the pitch differences in the query pattern:

Q When you did that, you knew that what was relevant was sublinearity with respect to the size of the dataset; right?
A That is always the assumption, that -- there are two notions of sublinearity which should not be confused.
And one notion, which is relevant to these proceedings, is sublinearity with respect to the size of the dataset. And then another notion, which occurs in Ghias in the references, is in comparison with brute-force search.
And I reiterate they are not the same

Moulin Depo. 158:14-159:4.

Q Did you ever read this and say to yourself, "What it says here about these algorithms, in this description of these algorithms, tells me that we have a sublinear time search"?
A No. I don't represent that, no.

Moulin Depo. 91:18-22.

Q -- were you trying to convey that these searches here in Ghias were sublinear with respect to the size of the dataset?
A No. No. The discussion there was relative to brute-force search. And it is only sublinear -- it is known to be sublinear in that sense. I do not know whether it would be sublinear in the size of the database for that particular algorithm.

Moulin Depo. 108:9-17

224. Petitioner's Declarant testified that when he wrote the following paragraph in his Declaration (Moulin Decl. ('237) ¶123):

> faster than "brute force" searches. Ex. 1010 at 6:23-35. In particular, Ghias discloses searches whose execution times are proportional to the logarithm of the size of the data set (*id.* at 6:24-28 ("O(kn) or O(nlog(m)")), which, as explained above in Section V(D), are sublinear (Ex. 1001 at 8:54-63).

he really meant sublinear with respect to the query, not the database or data set being searched:

> Q Is it the case that Ghias discloses a search whose execution time is proportional to the log of the size of the dataset being searched?
>
> A No. Okay. Again, if you say the dataset being searched, that would be N. Okay?
>
> What I intended there was the query dataset, which is M.
>
> Q So when you wrote "proportional to the log of the size of the dataset," you meant that to mean the query dataset?
>
> A Yes. Yes.
>
> Q Not the -- not the database?
>
> A That's -- that's correct. There's no database here. It's a problem of matching a query to a single song.

Moulin 103:1-15.

Q And by "logarithm of the dataset," you mean logarithm of the size of the query, not the dataset to be searched; is that true?
A I believe we discussed that earlier. Okay. Should have -- here it is the log, indeed, of the query dataset. So log(m).
Q So it should -- this should be interpreted -- when you write "dataset," you're -- what you mean is the query dataset, not the dataset to be searched?
A That's correct. So the logarithm of the dataset is log(m), which is written below. Yes.
Q Which is the query dataset?
A That's right.

Moulin 154:14-155:2

Q Why is it -- why did you think it was relevant to tell the Board that information? What would they do with it?
MR. ELACQUA: Calls for speculation. Objection.
THE WITNESS: I don't remember why I wrote it that way. Again, I agree it could be written in a more precise way. So it should have been the log of the query dataset. The word "query" should have appeared there.
BY MR. DOVEL:
Q Why don't you write that in so we don't have that confusion.
MR. ELACQUA: Again, I'm going to object.
THE WITNESS: I've said it. So the word "query" should be included before "dataset" in the box on line 4.
BY MR. DOVEL:
Q Why do you object to writing it in there?
A Why should I write it? I just said it.

Moulin Depo. 155:12-156:6.

A Because it's very clear. I'm saying the word "query" should have been incorporated before "dataset."

Moulin Depo. 156:22-157:3.

Well, one possible reason you wanted to give the Board this information is so that they would misread it and be misled.
That's a possibility; right?
A It's not at all the -- the reason. So it alone -- there are four documents. Some, they are, like, 90 pages each. Some of the words could have been better chosen. In particular, the word "query" should have been there. I have acknowledged that.
Q But assume --
A Again, I have acknowledged that this was not written the best way.

Moulin Depo. 157:6-18. I agree with Dr. Moulin—that the referenced passages in Ghias do not disclose a sublinear search with respect to the size of the dataset.

225. Petitioner fails to satisfy its burden of demonstrating that Ghias teaches a "sub-linear time search." As support for the "sub-linear" elements, Petitioner (and corresponding Declaration) exclusively rely on the statement addressed above—that Ghias discloses "searches whose execution times are proportional to the logarithm of the size of the data set" based on the disclosed running times of O(kn) or O(nlog(m)). Pet. ('237) at 41 (*quoting* Iwamura 6:23-35 and 6:24-28):

226. Petition:

> Independent claims 1 and 5 of the '237 patent further require that the search be "sublinear." Ex. 1001 at Claims 1, 5. Ghias discloses search algorithms that are substantially faster than "brute force" searches. Ex. 1010 at 6:23-35. In particular Ghias discloses searches whose execution times are proportional to the logarithm of the size of the data set (*id.* at 6:24-28 ("O(kn) or O(nlog(m)")), which, as explained above in Section V(D), are sublinear (Ex. 1001 at 8:54-63); Ex. 1004 at ¶ 123.

Pet. ('237) at 41.

227. Chart in Petition:

Claim 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Ghias determines the identification of a media work by "search[ing] the melody database" (2:50-59) to locate matching "sequence[s] of digitized representations of relative pitch differences," i.e., extracted features (Abstract). This is sublinear because execution time is proportional to the logarithm of the data set. 6:24-28 ("O(nlog(m)"). Ex. 1004 at ¶ 123. This identifies a list of neighbors, i.e., "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. ('237) at 42-43.

Claim 5(b.2):

| | |
|---|---|
| 2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and | Petitioner incorporates the above discussion of Ghias regarding Claim 1b. |

Pet. (‘237) at 44.

228. Declaration:

> 123. It is my opinion that Ghias further discloses the elements of claims 1 and 5 of the '237 patent that require that the search be "sublinear." Ex. 1001 at Claims 1, 5. In particular, Ghias discloses search algorithms that are substantially faster than "brute force" searches. Ex. 1010 at 6:23-35. In particular, Ghias discloses searches whose execution times are proportional to the logarithm of the size of the data set (*id.* at 6:24-28 ("O(kn) or O(nlog(m)")), which, as explained above in Section V(D), are sublinear (Ex. 1001 at 8:54-63).

Moulin Decl. (‘237) ¶123 (the paragraph addressed above).

229. Declaration Charts:

Claim 1(b):

| | |
|---|---|
| b) determining, by the computer system, an identification of the media work using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor; and | Ghias discloses determining the identification of a media work by "search[ing] the melody database" (2:50-59) to locate matching "sequence[s] of digitized representations of relative pitch differences," i.e., extracted features (Abstract). Ghias further discloses that this search is sublinear because its execution time may be proportional to the logarithm of the data set. 6:24-28 ("O(nlog(m)"). Ghias further discloses that this search identifies a list of neighbors, i.e., "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. (‘237) ¶127

Claim 5(b.2):

| | |
|---|---|
| 2) determining, by the computer system, an identification of the media work using the features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor, and | I incorporate my above discussion of Ghias regarding Claim 1b. |

Moulin Decl. ('237) ¶127.[27]

230. As I explained above in detail, this discussion and the passage from Ghias quoted in the Petition and Declaration exclusively address sub-linearity with respect the number of pitch differences in the query ("m"), not the "size of the string (song)" ("n") (Ghias, 6:23-28), much less the size of the data set being searched ("N"), as required by a sub-linear search in the context of the '237 patent. Accordingly, although each individual comparison can be more efficient using the searches disclosed in Ghias, the computational time it takes to search the database always grows linearly with the size of the dataset. As a result, the disclosed searches in Ghias are linear, not sublinear.

231: *Board's concerns*: I now addresses the Board's specific concerns (identified in its Decision in the '237 IPR) with respect to whether Ghias discloses the claimed "sub-linear time search." In instituting Ground 1, the Board

[27] Petitioner's expert confirmed that the other passages that he cites relating to other claim elements do not disclose a search that is sublinear. *See, e.g.*, Moulin Depo. 151:1-5; 151:6-12; 152:3-9 (addressing Ghias 2:50-59).

preliminary found that Ghias disclosed the claimed "sublinear time search" based on the disclosed "sub-linear approximate string matching" disclosed in Ghias:

> Additionally, as we found above, the claims do not specify that the sub-linear search must be performed on a subset of all of the records, and not information within individual records. As such, we are persuaded that the sub-linear approximate string matching, in Ghias, satisfies the claimed recitation of "using the received features extracted from the media work to perform a sub-linear time search of extracted features of identified media works to identify a neighbor."

Decision ('237) at 18-19.

232. As I demonstrated above, however, the "approximate string matching" algorithms disclosed in Ghias are only sub-linear with respect to the "m … the number of pitch differences in the query" not "n… the size of the string (song))" or with respect to N, the size of the dataset:

> 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).

Ghias, 6:23-28.

233. As I noted above, Petitioner's Declarant confirmed that Ghias, and the approximate string matching algorithms disclosed in Ghias, do not disclose a sub-linear search with respect to the size of the data set—the relevant inquiry in the

context of the '237 patent. Accordingly, the premise underlying the Board's preliminary finding—that the approximate string matching algorithms disclosed in Ghias have sub-linear properties with respect to the dataset—is wrong.[28]

### 2. approximate nearest neighbor search (claim elements 9(b) and 13(b.2)).

234. As I explained above, an "approximate nearest neighbor search" is a sublinear search identifying a close match that is not necessarily the closest match." Section V(D); Decision ('237) at 9.

235. One skilled in the art would understand that Ghias does not disclose the claimed "approximate nearest neighbor search" for at least two independent reasons.

236. Reason 1: One skilled in the art would understand that Ghias does not disclose an "approximate nearest neighbor search" because Ghias does not disclose "identifying a close match that is not necessarily the closest match."

237. To disclose an approximate nearest neighbor search, Ghias must disclose a search that does not necessarily find the closest match. *See* Section

---

[28] I note that Petitioner's Declarant also confirmed that searching a subset of information within individual records (*e.g.*, not looking at "every single character in the dataset") does not establish a sub-linear search. Moulin Depo. 37:18-38:5.

V(D). "A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point." '237, 9:12–19. A search that always (necessarily) identifies an exact or the closest match is not an approximate nearest neighbor search because a neighbor search identifies a "close, but not necessarily exact or closest, match." Section V(D); Decision ('237) at 8.

238. Ghias discloses a search algorithm that necessarily finds the closest match. Ghias does not expressly disclose a search that does not necessarily identify an exact or closest match. And one skilled in the art would understand that such a search is not inherent (necessarily present) in Ghias.

239. 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).

Petitioner's Declarant confirmed, consistent with my understanding, that Ghias teaches these three potential outputs:

> Would you agree that Ghias teaches sort of three options? We've got -- it's going to produce the exact match; it's going to produce the most approximate matching; or it's going to produce this ranked list?
>
> A Yes.

Moulin Depo. 341:16-21.

240. For all three outputs, Ghias always identifies an exact or the closest match:

241. (1) exact match: If the search produces an exact match, it necessary produces an "exact or closest, match" and therefore does not disclose an "approximate nearest neighbor search." Petitioner's Declarant agreed with my understanding:

> If it produces the exact matching melody, would you agree that Ghias teaches providing or identifying the closest match?
>
> A Yes.

Moulin Depo. 341:23-342:1.

242. (2) ranked list: If the search produces a ranked list, it necessarily identifies as part of the ranked list either an exact match (if there is one) or the closest match—*i.e.*, the top ranked match—and therefore does not disclose an "approximate nearest neighbor search" that does not necessarily identify an exact or the closest match. At the top of the ranked list (*i.e.*, the number 1 ranked match

in the list) is an exact or the closest match. For example, assume for illustrative purposes that the work to be identified is 500. Assume that the list outputs in ranked order:

1st closest: 502
2nd closest: 510
3rd closest: 530 and
4th closest: 570.

The ranked list identified the closest match as 502. The closest match will never be excluded from the list of matches returned. As another example, assume that the list outputs in ranked order:

1st closest: 500
2nd closest: 510
3rd closest: 530 and
4th closest: 570.

In this example, the ranked list identified an exact match as 500. The exact match will never be excluded from the list of matches returned.[29] Accordingly, this

---

[29] The list of matches within a given error-tolerance includes the full list of matches except those matches outside a given error-tolerance. Because the closest match is among the matches retrieved from the database, and the closest match is

approach necessarily identifies the closest match and therefore is not an "approximate nearest neighbor search."

243. Petitioner's Declarant confirmed, consistent with my understanding, that the ranked list approach identifies the closest match:

> Would you agree that for the embodiment in Ghias that produces a ranked list, that the list is going to identify the match that has been deemed by the algorithm to be the closest match?
>
> A Yes.
>
> Q It doesn't merely say, "Here's a set of matches that are close"; it's going to identify the one that it has deemed to be the closest match?
>
> A So in the embodiment where it outputs a list, you are saying?
>
> Q Yes.
>
> A So it will output the list of what it believes or deems are the closest matches.
>
> Q And on that list, it's also going to identify the one that is the closest match?
>
> A That is deemed by the algorithm to be -- by the approximate algorithm to be the closest match.
>
> Q Yes?
>
> A Yes.

Moulin Depo. 356: 2-21.

---

not subsequently excluded from that list, the closest match will always be among the list of matches returned.

244. (3) single most approximate matching melody: If the search identifies the single most approximate matching melody, it necessarily identifies the closet match and is therefore not the claimed "approximate nearest neighbor search." Petitioner's Declarant agreed with my understanding:

Q If it returns -- if it identifies anything as a -- as a match, it's going to be identifying the closest possible match; right?
A Yes.

Moulin Depo. 345:16-346:11.

245. Petitioner's expert confirmed that for all three outputs, Ghias teaches a system that will always (necessarily) identify the closest match. Moulin Depo. 352:22-353:2. Accordingly, for all three potential outputs, Ghias necessarily identifies an exact or the closest match. Ghias does not disclose an "approximate nearest neighbor search" which identifies "a close, but not necessarily exact or closest, match."

246. Reason 2: Ghias does not disclose an "approximate nearest neighbor search" because Ghias does not disclose a sublinear search. As I explained above, an "approximate nearest neighbor search" is "one example" of a "sublinear search." Section V(D). Also, as I demonstrated above, Ghias does not disclose a "sublinear search." Section VI(B)(1). Accordingly, Ghias does not disclose the claimed "approximate nearest neighbor search."