

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google Inc.
Petitioner,
v.
Network-1 Technologies, Inc.
Patent Owner.

Case IPR2015-00345
U.S. Patent 8,205,237

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google Inc.
Petitioner,
v.
Network-1 Technologies, Inc.
Patent Owner.

Case IPR2015-00347
U.S. Patent 8,010,988

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google Inc.
Petitioner,
v.
Network-1 Technologies, Inc.
Patent Owner.

Case IPR2015-00343
U.S. Patent 8,640,179

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google Inc.
Petitioner,
v.
Network-1 Technologies, Inc.
Patent Owner.

Case IPR2015-00348
U.S. Patent 8,656,441

Patent Owner Network-1's Disclosure of Demonstrative Slides

Iwamura

- '237 Ground 1
- '237 Ground 3
- '988 Ground 3

Ghias

- '237 Ground 2
- '988 Ground 1
- '988 Ground 2
- '179 Ground 2
- '441 Ground 2

Conwell

- '179 Ground 1
- '441 Ground 1

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
'988	33	sublinear approximate nearest neighbor
	15	non-exhaustive search identifying a neighbor

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11
'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

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

Moulin (A2006-Part 1) 82:23-83:3

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11

'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

Petitioner's expert confirmed Boyer-Moore algorithm is linear:

sub-linear time search: "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched"

Board's construction

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

Moulin (A2006-Part 1) 61:18-24

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11

'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

No reasonable explanation for sublinear statements

sub-linear time search: "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched"

Board's construction

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

Moulin (A2006-Part 1) 74:20-24

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11

'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

No reasonable explanation for sublinear statements

sub-linear time search: "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched"

Board's construction

16 Q Okay. Would you agree, sir, that if --
17 that **one way to read this would be that you were**
18 **claiming** that the claim language, "perform a
19 **sublinear time search,**" was satisfied by searching
20 using the **Boyer-Moore algorithm?**
21 A That **might be one way of reading it.** It's
22 **not the way I'm reading this** now.

Moulin (A2006-Part 1) 78:16-22

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11

'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

No reasonable explanation for sublinear statements

sub-linear time search: "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched"

Board's construction

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

Moulin (A2006-Part 1) 75:23-76:3

Petition asserts that Boyer-Moore algorithm is sublinear:

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.

'237 Petition 10-11
'237 Moulin Decl. (1004) ¶75

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.

'237, Moulin Decl. (1004) ¶ 72

No reasonable explanation for sublinear statements

sub-linear time search: "a search whose execution time scales with a less than linear relationship to the size of the data set to be searched"

Board's construction

19 Q To be **candid with the Board**, wouldn't it
20 have been better to say, "Board, the Boyer-Moore
21 algorithm is **linear, not sublinear**"?
22 A Listen, there are many words that I -- I'm
23 sure **I could have chosen better words**. So I agree
24 with you, there is **probably better ways to write**
25 this. **I don't dispute that**.

Moulin (A2006-Part 1) 79:19-25

New Reply Theory:

Due to the foregoing characteristics, Iwamura discloses a sublinear search when .wav files are added to a database of MIDI files.³ For instance, assume a

'237 Reply at 17

I. Petitioner Reply to Patent Owner Response and Patent Owner Reply to Opposition To Amend

A reply may only respond to arguments raised in the corresponding opposition. § 42.23. While replies can help crystalize issues for decision, a reply that raises a new issue or belatedly presents evidence will not be considered and may be returned. The Board will not attempt to sort proper from improper portions of the reply. Examples of indications that a new issue has been raised in a reply include new evidence necessary to make out a *prima facie* case for the patentability or unpatentability of an original or proposed substitute claim, and new evidence that could have been presented in a prior filing.

New Reply Theory:

A reply may always raise responsive arguments.

'237 Reply at 19

Patent owner's argument:

Boyer-Moore algorithm, however, does not disclose a sublinear search.

'237 Response at 12

Responsive argument in reply:

“Patent owner's is wrong because . . . ”

Patent owner's argument:

Boyer-Moore algorithm, however, does not disclose a sublinear search.

'237 Response at 12

New Reply issue:

Due to the foregoing characteristics, Iwamura discloses a sublinear search when .wav files are added to a database of MIDI files.³ For instance, assume a

'237 Reply at 17

New Reply Theory:

Due to the foregoing characteristics, Iwamura discloses a sublinear search when .wav files are added to a database of MIDI files.³ For instance, assume a

'237 Reply at 17

Iwamura stores reference songs in two different file formats—low resolution MIDI, and high resolution waveform (".wav")—each requiring a different amount of disk space to represent the same music. MIDI files are "compressed" and

'237 Reply at 16

Evidence?

Evidence?

higher resolution works are added to a reference database of lower resolution works.

'237 Reply at 16

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

non-exhaustive search: “a search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches.”

Petitioner’s construction

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

Reference melodies are possible matches

compared. The **reference melody** that gives the least difference **is returned as a search result.**

Iwamura (1012) 7:54-55

- 12 Q When we do the search that's described in
13 **Iwamura**, is the -- are the -- the **possible matches**
14 the set of the **melodies in the database** in Iwamura?
15 A **Yeah**, that you try to **match melodies**.

Moulin (A2006 Part 3) 206:12-15

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

Reference melodies are possible matches

compared. The reference melody that gives the least difference is returned as a search result.

Iwamura (1012) 7:54-55

Iwamura compares all possible matches

22 Q Now, let's return to Iwamura. The --
23 we've talked about how the possible matches that can
24 be returned in Iwamura are musical works that are in
25 the reference database; right?

1 A Yes.

2 Q You would agree that in Iwamura, the
3 search that's identified there does make a
4 comparison to each of the possible musical works
5 that could be returned as a match?

6 MR. ELACQUA: Objection.

7 THE WITNESS: To each of the musical works,
8 yes.

Moulin (A2006 Part 3) 222:22-223:8

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

Reference melodies are possible matches

compared. The reference melody that gives the least difference is returned as a search result.

Iwamura (1012) 7:54-55

Iwamura compares all possible matches

15 Q But it does examine each of the possible
16 musical works -- or each of the musical works that
17 could be returned as a possible match?
18 A Yes.

Moulin (A2006-Part 3) 217:15-18

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

Moulin (A2006-Part 3) 247:18-20

Reply quoting Iwamura 9:9-11

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

'237 Reply at 7

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:8-11

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Database records consist of pitch values

Record in the database

										
5	-1	1	4	3	5	0	-1	-1	-2	-2

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Computes absolute difference

Record in the database											
	5	-1	1	4	3	5	0	-1	-1	-2	-2
Work to be identified											
	4	3	5	0	-1	-2	-2	5	-10	2	
Computation (absolute difference)			0	1	2	6	5	0	10	3	Total: 27

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Option 1: shift note by note

Record in the database											
	5	-1	1	4	3	5	0	-1	-1	-2	-2
Work to be identified											
	4	3	5	0	-1	-2	-2	5	-10	2	
Computation (absolute difference)	2	6	1	5	5	7	5	9	3		

shifted 1 note

Total: 43

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Option 2: shift to next peak note (Iwamura approach)

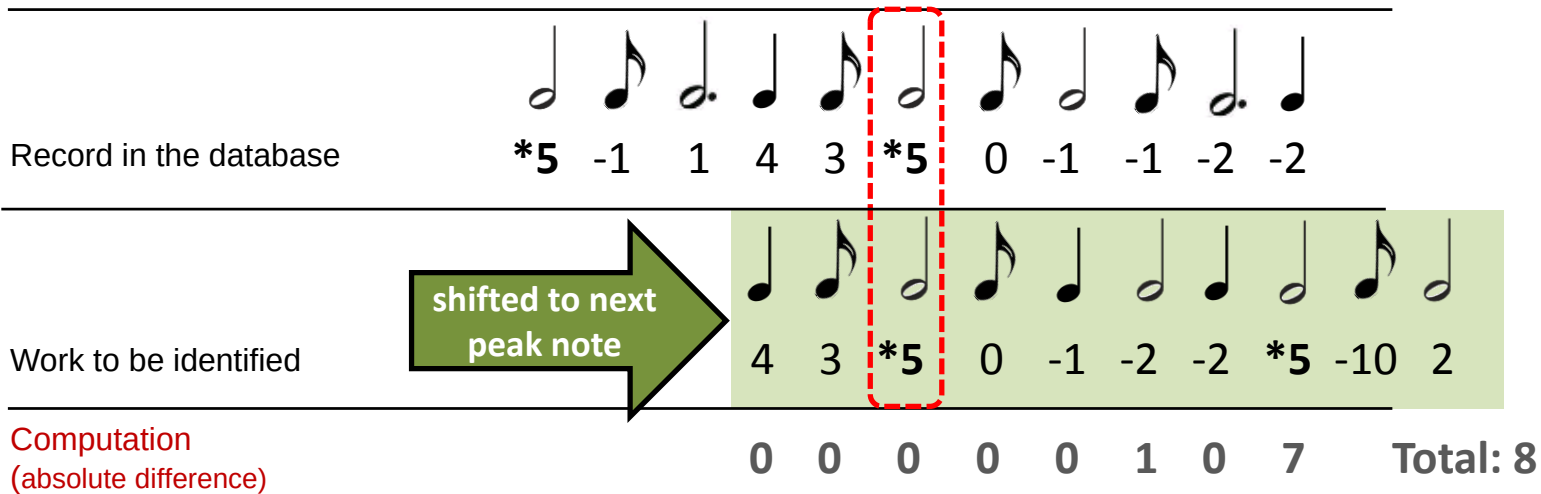
Record in the database												
	*5	-1	1	4	3	*5	0	-1	-1	-2	-2	
Work to be identified												
	4	3	*5	0	-1	-2	-2	*5	-10	2		
Computation (absolute difference)			0	1	2	6	5	0	10	3	Total: 27	

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Option 2: shift to next peak note (Iwamura approach)



avoided four calculations

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

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

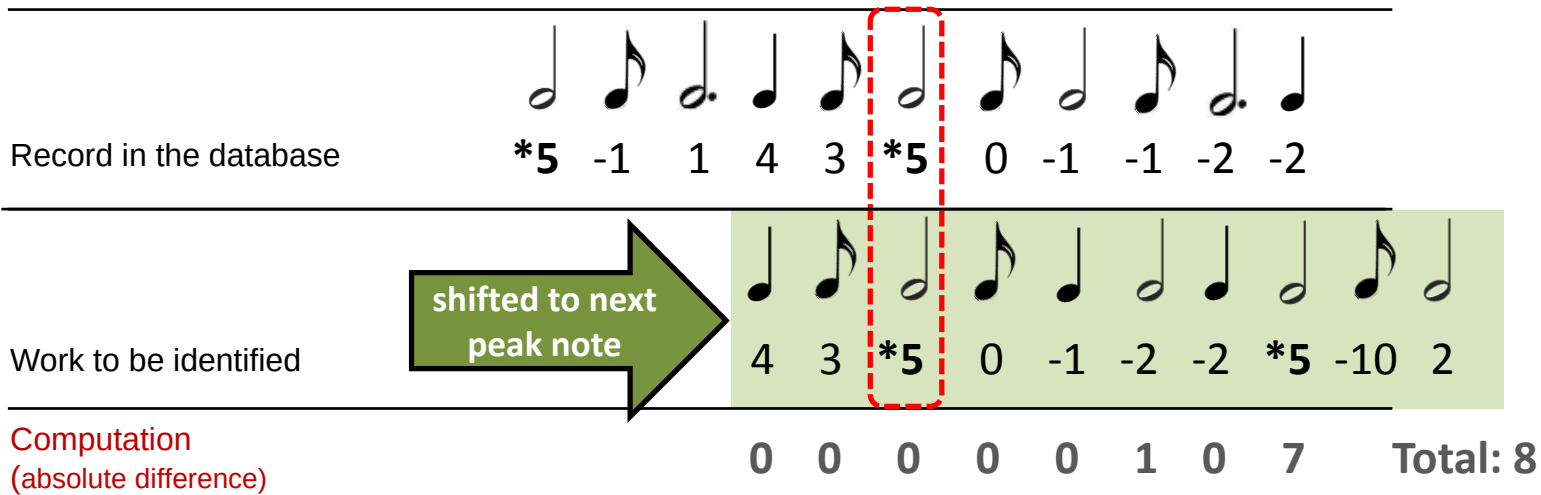
Moulin (A2006-Part 3) 247:18-20

Actual quote from Iwamura 9:8-11

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

Iwamura (1012) 9:9-11

Option 2: shift to next peak note (Iwamura approach)



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

Iwamura (1012) 9:9-11

nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

non-exhaustive search: “a search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches.”

Petitioner’s construction

Board:

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

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57

Board:

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

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57

Record in the database												
	*5	-1	1	4	3	*5	0	-1	-1	-2	-2	
Work to be identified												
	4	3	*5	0	-1	-2	-2	*5	-10	2		
Computation (absolute difference)	0										Total: 0	limit = 20

Board:

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

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57

Record in the database											
	*5	-1	1	4	3	*5	0	-1	-1	-2	-2
Work to be identified											
	4	3	*5	0	-1	-2	-2	*5	-10	2	
Computation (absolute difference)	0	1									
Total: 1											limit = 20

Board:

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

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57

Record in the database											
	*5	-1	1	4	3	*5	0	-1	-1	-2	-2
Work to be identified											
	4	3	*5	0	-1	-2	-2	*5	-10	2	
Computation (absolute difference)	0	1	2								
Total: 3											limit = 20

Board:

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

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57

Record in the database													
	*5	-1	1	4	3	*5	0	-1	-1	-2	-2		
Work to be identified													
	4	3	*5	0	-1	-2	-2	*5	10	2			
Computation (absolute difference)	0	1	2	6	5	0	10	Total: 24					limit = 20

Board:

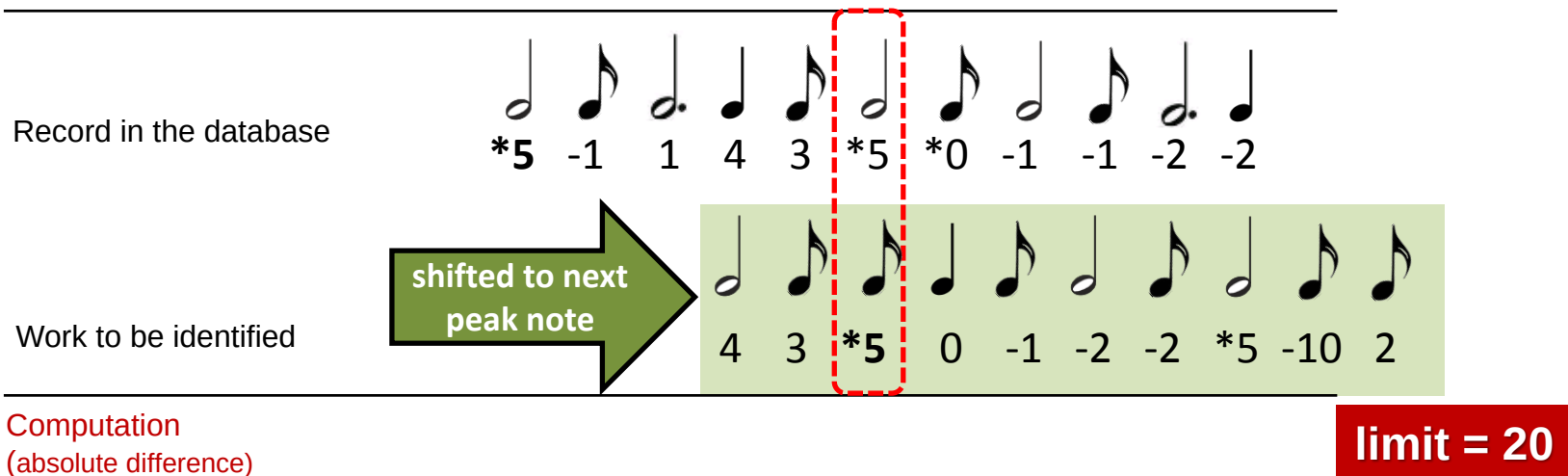
18. We do not agree. If, in Iwamura, the computational limit is reached, the search is stopped, even if not all of the records have been searched. Per our

'237 Decision at 11-12

Iwamura:

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

Iwamura (1012) 7:56-57



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

2 A Yes. Yes.

Board:

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

'237 Decision at 11-12

Iwamura:

Iwamura compares all possible matches

result window 6 in FIG. 1. In this manner, the entered melody is shifted to each peak in each reference melody and compared. The reference melody that gives the least difference is returned as a search result.

Iwamura (1012) 6:51-55

15 Q But it does examine each of the possible
16 musical works -- or each of the musical works that
17 could be returned as a possible match?
18 A Yes.

Moulin (A2006-Part 3) 217:1-18

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

	Independent claim	Key missing elements
'237	1	sub-linear-time search identify a neighbor
	5	sub-linear-time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

Approximate nearest neighbor

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. Often such an algorithm will find the nearest

Ex. (A2008) at 5

with some specified probability may be used. One example of a sub-linear time search is an approximate nearest neighbor search. A nearest neighbor search always finds the closest point to the query. An approximate nearest neighbor search does not always find the closest point to the query. For example, it might do so with some probability, or it might provide any point within some small distance of the closest point.

'237, 9:12-19

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

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).

'237 Petition at 12

skipping unimportant portions always necessarily identifies the closest match

14 Is it the case that if we do the Iwamura
15 search using the peaks as our basis, and we set up
16 our database such that **the unimportant peaks are**
17 **skipped**, that **we're still going to be identifying**
18 **the closest match** when we produce our results?

19 A That would be assuming that no peaks have
20 been dropped and everything we discussed yesterday.
21 **Dropping an unimportant part is not going to affect**
22 **the ability to find the best match.**

Moulin (A2006-Part 4) 317:14-22

Petitioner's new Reply theory:

1012, 7:52–54; Paper No. 17 at 29–30; Ex. 2005, ¶¶ 162–63. If the closest matching melody segment does not occur at a peak alignment, Iwamura will neither consider nor locate it. See Ex. 1020 at 128:13–135:14, Exh. 9.

'237 Reply at 14

Query: *5 4 3 2 1
Reference: *6 2 2 2 1 *6 5 4 3 2 1

Comparisons:

	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8	Position 9	Position 10	Position 11	Position 12	Total Absolute Difference
Comparison No. 1	1	-	-	-	-	-	-	-	-	-	-	-	5
Comparison No. 2	2	1	-	-	-	-	-	-	-	-	-	-	5
Comparison No. 3	3	2	1	-	-	-	-	-	-	-	-	-	4
Comparison No. 4	4	3	2	1	-	-	-	-	-	-	-	-	4
Comparison No. 5	*5	4	3	2	1	-	-	-	-	-	-	-	5
Comparison No. 6	*5	4	3	2	1	-	-	-	-	-	-	-	6
Comparison No. 7	-	*5	4	3	2	1	-	-	-	-	-	-	12
Comparison No. 8	-	-	*5	4	3	2	1	-	-	-	-	-	15
Comparison No. 9	-	-	-	*5	4	3	2	1	-	-	-	-	15
Comparison No. 10	-	-	-	-	*5	4	3	2	1	-	-	-	12
Comparison No. 11	-	-	-	-	-	*5	4	3	2	1	-	-	5
Comparison No. 12	-	-	-	-	-	-	*5	4	3	2	1	-	0
Comparison No. 13	-	-	-	-	-	-	-	*5	4	3	2	1	4
Comparison No. 14	-	-	-	-	-	-	-	-	*5	4	3	2	6
Comparison No. 15	-	-	-	-	-	-	-	-	-	*5	4	3	6
Comparison No. 16	-	-	-	-	-	-	-	-	-	-	*5	4	4

Reference: *6 2 2 2 1 *6 5 4 3 2 1

Aligning Peak Notes

'237 Reply at 12

US006188010B1

(12) United States Patent
Iwamura

(10) Patent No.: US 6,188,010 B1
(45) Date of Patent: Feb. 13, 2001

(51) MUSIC SEARCH BY MELODY INPUT
(75) Invention: Ryoichi Iwamura, San Diego, CA (US)
(73) Assignees: Sony Corporation, Tokyo (JP); Sony Electronics, Inc., Park Ridge, NJ (US)

6,001,174 * 2/2000 Takahashi 6,609
OTHER PUBLICATIONS
MIDISOFT Desktop Sheet Music 2.0 and MIDISOFT Studio 6.0 <http://www.midssoft.com/vgrade.htm> Apr. 10, 1999.
* cited by examiner
Primary Examiner—Stanley J. Witkowski
(73) Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman LLP

0429,268
Feb. 28, 1999

001B 1594, GI01 1/25
84/609, 84/477 R
84/609-614, 614-636, 84/477 R, 478, 484/307 A

References Cited
PATENT DOCUMENTS
1995 Nakamura et al. 84/609
1997 Grosse et al. 84/609 X
1998 Winsky et al. 84/609 X
1999 Cantata 84/609
1999 Montyana 84/609
1999 Miller et al. 84/609
1999 Looney et al. 84/609
1999 Catta 84/609

(57) ABSTRACT
A method to enable one to search for a song title when only its melody is known. An interface is used which allows the user to enter the melody in an easy to understand manner. A remote music database with melody information is searched for the melody entered by the user using, for example, a peak or differential matching algorithm. Upon receiving a search request, a database server, i.e., a remote computer, which can be accessed by the user via a client PC, sends a web page, for example, in HTML or Java containing the search results to the client PC. The client PC receives and displays the search results on the monitor and exchanges data with the server. Sound may also be played through the sound card to connected speakers.

40 Claims, 8 Drawing Sheets

SEARCH WINDOW
Melody Search
PLAY CLEAR SEARCH
Search Result
Trautner R. Schumann

Google Ex. 1012

Petitioner's new Reply theory:

1012, 7:52–54; Paper No. 17 at 29–30; Ex. 2005, ¶¶ 162–63. If the closest matching melody segment does not occur at a peak alignment, Iwamura will neither consider nor locate it. See Ex. 1020 at 128:13–135:14, Exh. 9.

'237 Reply at 14

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

Iwamura always finds the closest match

of . . . media works" Ex. 1001 at Claims 1, 5, 9, 13, 25, 33. In Iwamura, the "search engine will find the closest melody from the database." Ex. 1012 at 9:24-25. For

'237 Petition at 8

Petitioner's new Reply theory:

1012, 7:52–54; Paper No. 17 at 29–30; Ex. 2005, ¶¶ 162–63. If the closest matching melody segment does not occur at a peak alignment, Iwamura will neither consider nor locate it. *See* Ex. 1020 at 128:13–135:14, Exh. 9.

'237 Reply at 14

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

“closest match” is based on criterion stated in Iwamura

22 Q And if it -- it finds one that's an exact
23 match, it will produce -- it will produce that;
24 right?

25 A Yes.

1 Q If it doesn't find one that's -- that's an
2 exact match, it's going to produce the one that has
3 the next best match; right?

4 A According to the approximate criterion
5 only.

6 So, again, "match" was defined with
7 respect to all the notes. Okay? There's a
8 criterion that's stated in the Iwamura; it is the
9 least absolute difference. And then there are
10 various approximations that are used, and it will
11 output the best match it finds using that
12 approximate criterion.

	Independent claim	Key missing elements
'237	1	sub-linear time search identify a neighbor
	5	sub-linear time search identify a neighbor
	9	approximate nearest neighbor
	13	approximate nearest neighbor
	25	identify a neighbor non-exhaustive search
	33	sublinear approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor

Iwamura

- '237 Ground 1
- '237 Ground 3
- '988 Ground 3

Ghias

- '237 Ground 2
- '988 Ground 1
- '988 Ground 2
- '179 Ground 2
- '441 Ground 2

Conwell

- '179 Ground 1
- '441 Ground 1

		Independent claim	Key missing elements
'237	{	1	sub-linear time search
		5	sub-linear time search
		9	approximate nearest neighbor
		13	approximate nearest neighbor
'988	{	15	non-exhaustive search search identifying a neighbor determining an action based on the identification
		1	non-exhaustive search neighbor search
'179 & '441	{	13	non-exhaustive search neighbor search
		25	non-exhaustive search neighbor search

Claims 1-3, 5-7?



IV. GROUND B: GHIAS ANTICIPATES CLAIMS 9–11, 13–15, AND 23–24

The Board instituted review on the ground that Ghias anticipates claims 9–11, **13–15**, and 23–24 (independent claims bolded). The parties dispute whether

'237 Reply at 20

The parties dispute whether Iwamura and Ghias disclose one element—(1)

"approximate nearest neighbor search"—and whether Iwamura discloses two

'237 Reply at 1

sub-linear time search: “a search whose execution time scales with a less than linear relationship to the size of the data set to be searched”

Board's construction

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~~ **query** (*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).

‘237, Moulin Decl. (1004) ¶ 123

6 Well, one possible reason you wanted to
7 give the Board this information is so that they
8 would misread it and be misled.
9 That's a possibility; right?
10 A It's not at all the -- the reason. So
11 it alone -- there are four documents. Some, they
12 are, like, 90 pages each. Some of the words could
13 have been better chosen. In particular, the word
14 "query" should have been there. I have acknowledged
15 that.

Moulin (A2006- Part 2) 157:6-15

		Independent claim	Key missing elements
'237	{	1	sub-linear-time search
		5	sub-linear-time search
		9	approximate nearest neighbor
		13	approximate nearest neighbor
'988	{	15	non-exhaustive search search identifying a neighbor determining an action based on the identification
		1	non-exhaustive search neighbor search
'179 & '441	{	13	non-exhaustive search neighbor search
		25	non-exhaustive search neighbor search

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

non-exhaustive search: “a search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches.”

Petitioner’s construction

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

Patent Owner's Expert:

266. One skilled in the art would understand that Ghias teaches an exhaustive search that compares the work to be identified (user input 23) with “all the songs” in the database—*i.e.*, “all possible matches.” One skilled in the art

Karypis Decl. (A2005-Part 3) ¶266

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

Karypis Decl. (A2005-Part 3) ¶289

Petitioner's Expert:

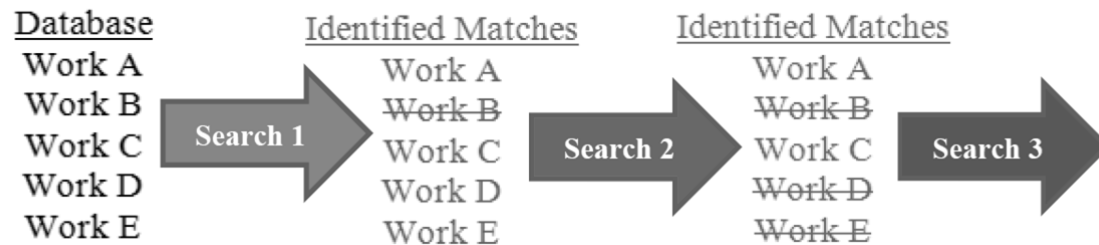
8 Q Would it be the case that the Ghias search
9 is going to be performing a comparison to each of
10 the melodies that are possible matches in the
11 database?
12 A It does a comparison, yes, to each of
13 them.

Moulin (A2006-Part 4) 323:8-13

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

subsequent search can be further refined with a new query, which in turn will display a more refined "list consisting of songs just retrieved." Ex. 1010, 7:5–8.



'179 Reply at 13

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

Database

Work A

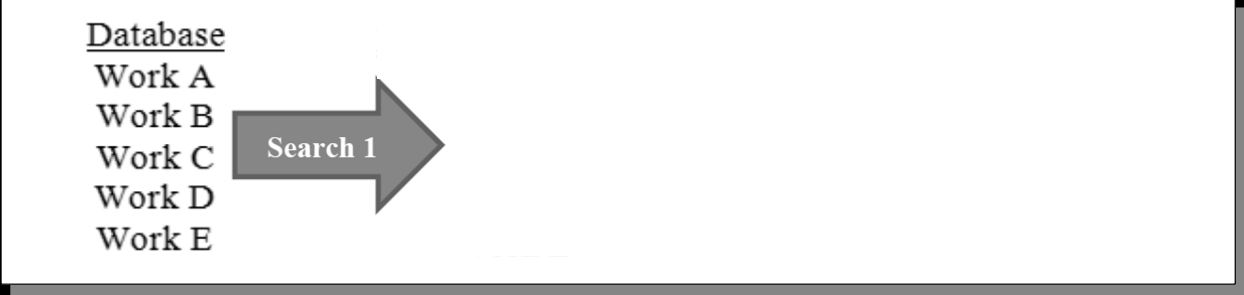
Work B

Work C

Work D

Work E

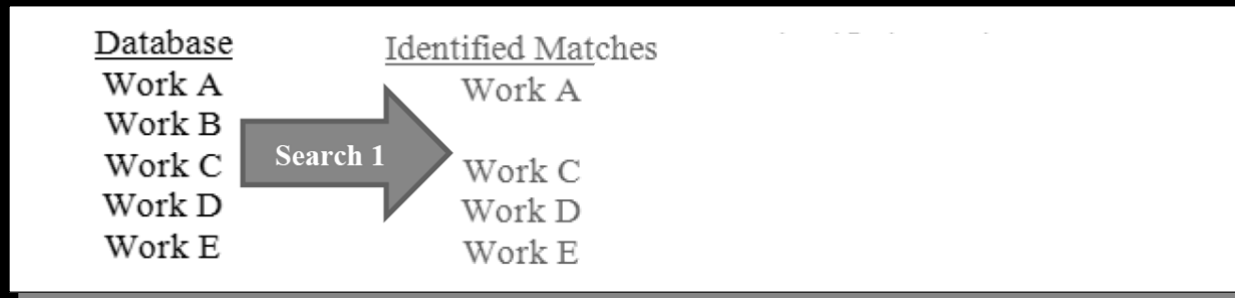
Search 1



“My Way”
chorus

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

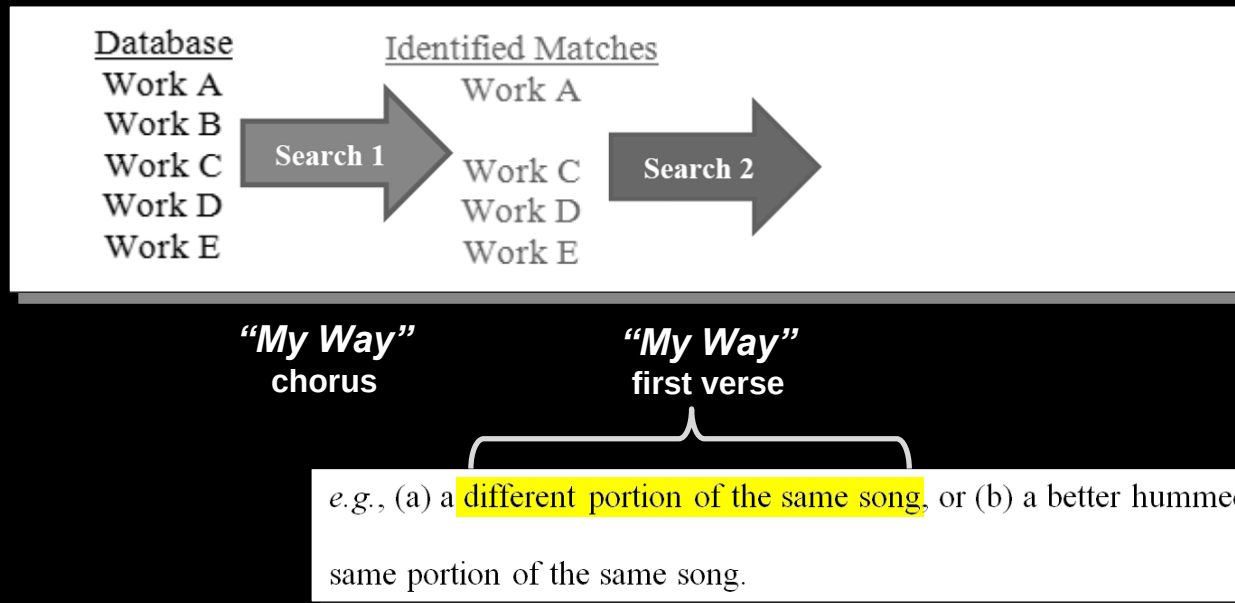
Board's construction



“My Way”
chorus

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

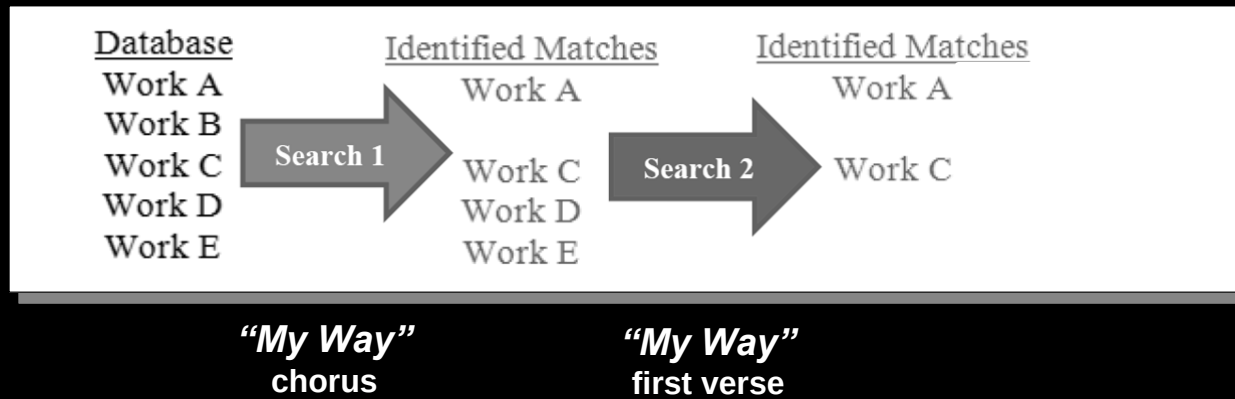
Board’s construction



Karypis Decl. (A2005-Part 3) ¶287

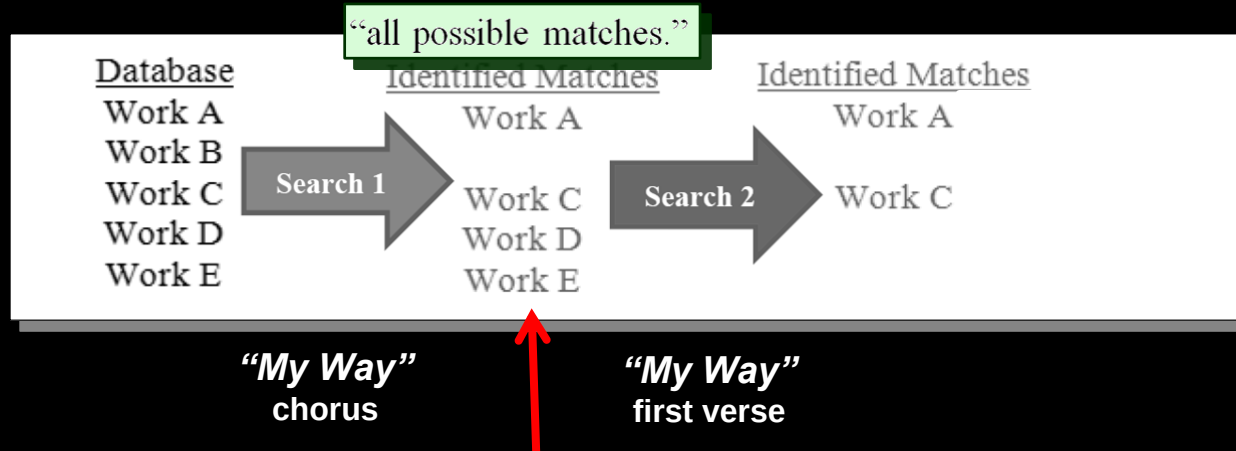
non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction



non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction



From the results of the query the user can identify the song of interest. If the list is too large, the user can perform a new query on a restricted search list consisting of songs just retrieved. This allows the user to identify sets of songs

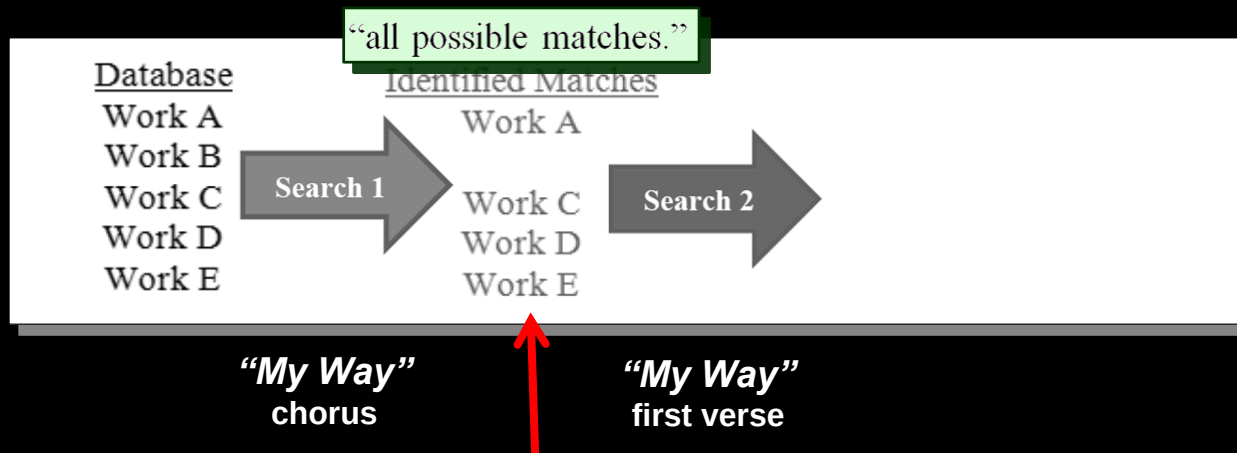
Ghias (1010) 7:4-9

the “restricted search list” comprises “all possible matches.”

Karypis Decl. (A2005-Part 3) ¶288

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction



13 Q Then we get a list that comes from there
14 of a set of our best matches?

15 A Deemed best matches, yes.

16 Q And then it's going to do another search
17 on those; right?

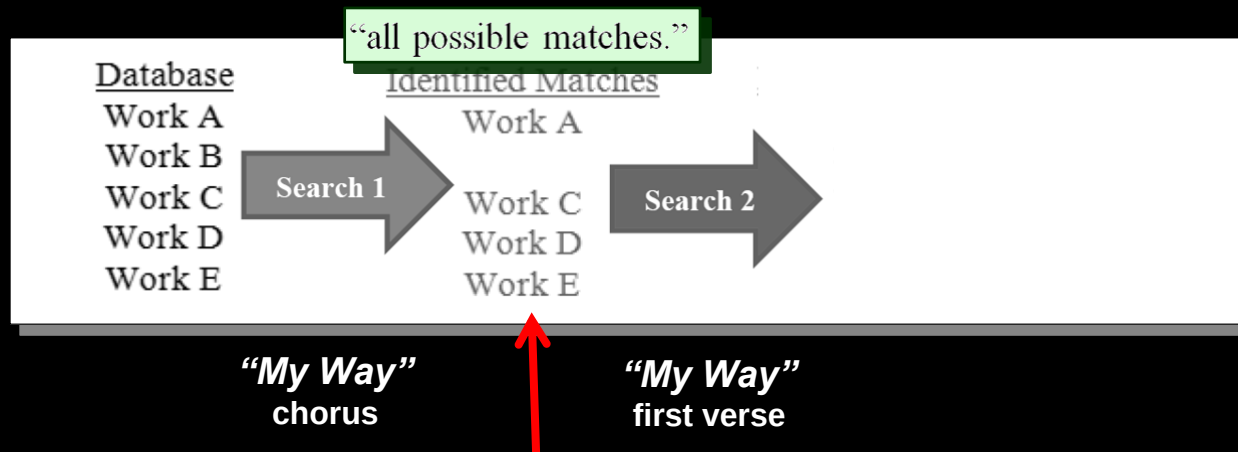
18 A On that list, yes.

19 Q And the possible matches for that search
20 are defined to be only the ones that are on the
21 list; is that right?

22 A That's correct.

non-exhaustive search: “a search that locates a match **without a comparison of all possible matches**”

Board's construction

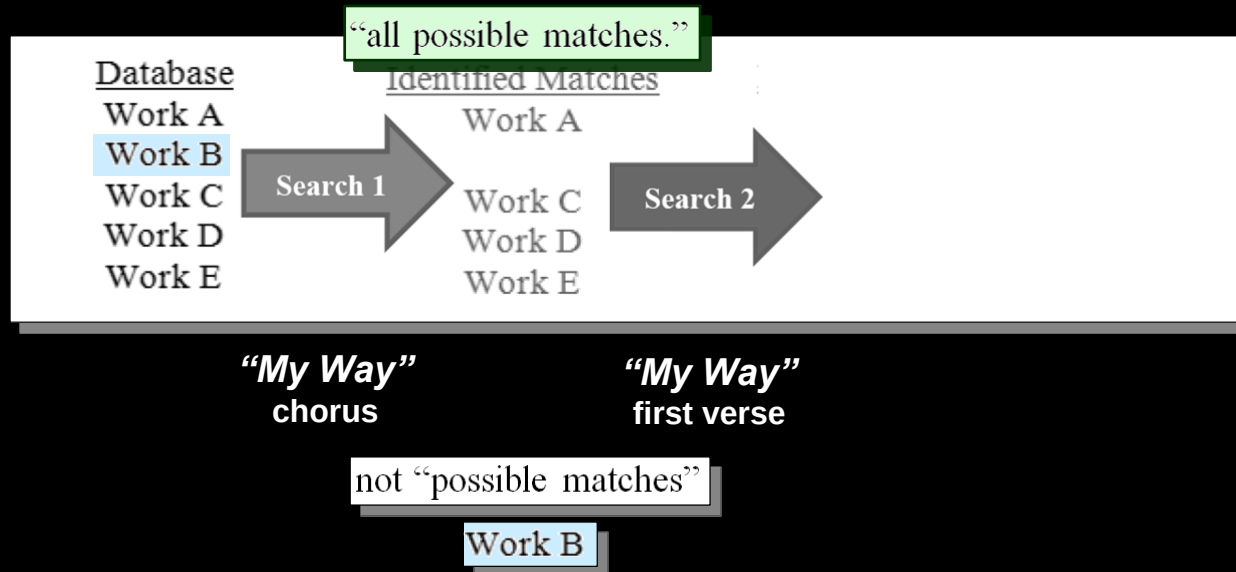


- 13 Q All right. What you're referring to here
14 is we now have a -- a list -- we do a search, and it
15 **results in a list of possible matches;** right?
16 A **Yes.**
17 Q Would you agree, then, that each of the --
18 it's done a comparison, and what it's identified --
19 that are **on the list are all possible matches;**
20 right?
21 A **Yes.**

Moulin (A2006-Part 4) 325:13-21

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

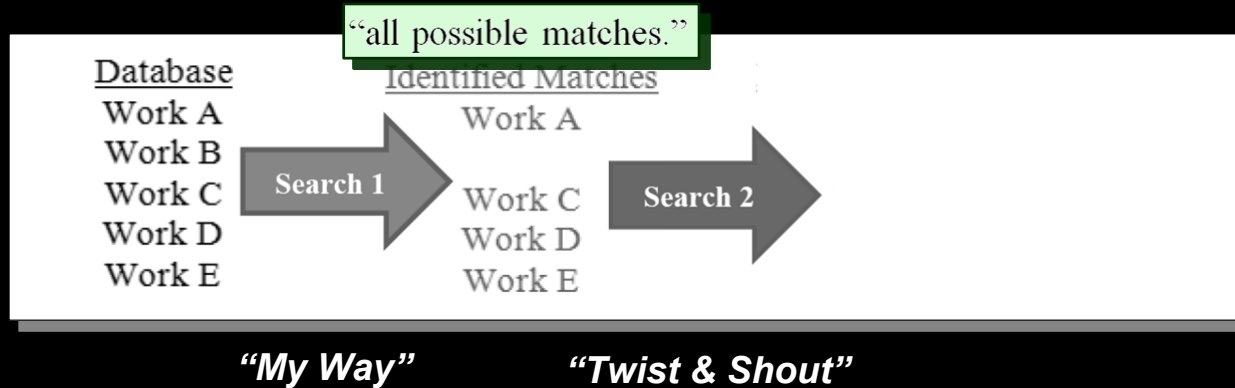
Board’s construction



The records that are not on the restricted list (i.e., in the blue dataset but not the green dataset) are not “possible matches” for the restricted search. The first search excludes from the list of ranked songs those songs that are not possible matches such that the “restricted search list” comprises “all possible matches.” Moulin

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

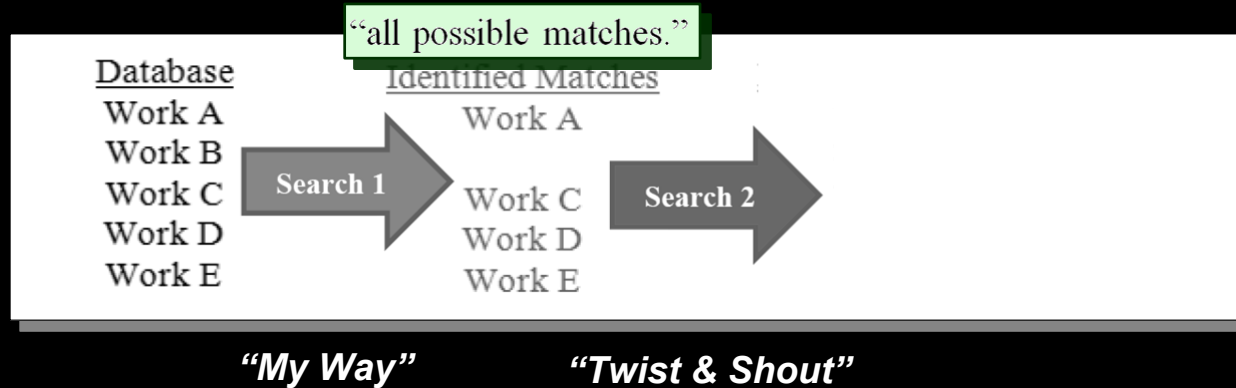


that would be the closest match to the second work. For instance, if Ghias ran a first search to identify Frank Sinatra's *My Way*, it would likely exclude The Beatles' *Twist and Shout* from the list of matches. If Ghias subsequently ran a second search to identify *Twist and Shout*, that search would consider only the results of the first search, which excluded *Twist and Shout*. The second search would not

'179 Reply at 16-17

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

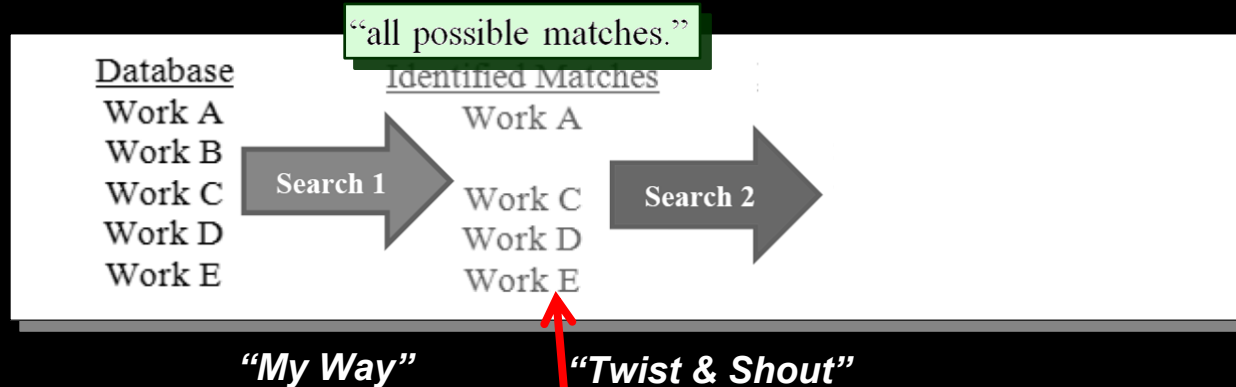


From the results of the query the user can identify the song of interest. If the list is too large, the user can perform a new query on a restricted search list consisting of songs just retrieved. This allows the user to identify sets of songs

Ghias (1010) 7:4-8

non-exhaustive search: “a search that locates a match **without a comparison of all possible matches**”

Board's construction



- 13 Q All right. What you're referring to here
- 14 is we now have a -- a list -- we do a search, and it
- 15 **results in a list of possible matches;** right?
- 16 A **Yes.**
- 17 Q Would you agree, then, that each of the --
- 18 it's done a comparison, and what it's identified --
- 19 that are **on the list are all possible matches;**
- 20 right?
- 21 A **Yes.**

Petitioner's new Reply theory:

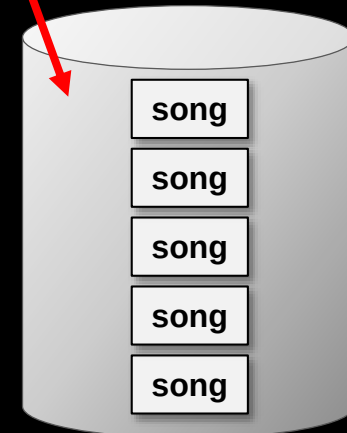
of the full database—as non-exhaustive. Ghias discloses string matching algorithms that operate faster than a brute force search. Paper No. 1 at 9–10 (citing Ex. 1010, 6:23–28 ("Running times have ranged from $O(mn)$ for the brute force algorithm to" substantially faster.) (emphasis modified)). While these algorithms

'988 Reply at 7

non-exhaustive search: "a search that locates a match without a comparison of all possible matches"

Board's construction

database



Petitioner's new Reply theory:

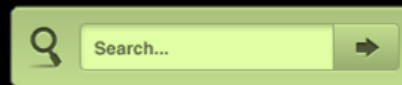
of the full database—as non-exhaustive. Ghias discloses string matching algorithms that operate faster than a brute force search. Paper No. 1 at 9–10 (citing Ex. 1010, 6:23–28 ("Running times have ranged from $O(mn)$ for the brute force algorithm to" substantially faster.) (emphasis modified)). While these algorithms

'988 Reply at 7

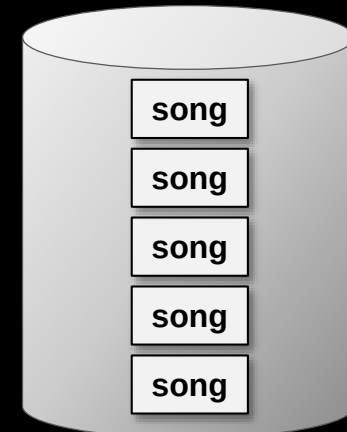
Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from $O(mn)$ for the brute force algorithm to $O(kn)$ or $O(n \log(m))$, where "O" means "on the order of," m is the number of pitch differences in the query, and n is the size of the string (song). See Ricardo Baeza-Yates and G. H.

Ghias (1010) 6:23-28

query



database



Petitioner's new Reply theory:

of the full database—as non-exhaustive. Ghias discloses string matching algorithms that operate faster than a brute force search. Paper No. 1 at 9–10 (citing Ex. 1010, 6:23–28 ("Running times have ranged from $O(mn)$ for the brute force algorithm to" substantially faster.) (emphasis modified)). While these algorithms

'988 Reply at 7

Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from $O(mn)$ for the brute force algorithm to $O(kn)$ or $O(n\log(m))$, where "O" means "on the order of," m is the number of pitch differences in the query, and n is the size of the string (song). See Ricardo Baeza-Yates and G. H.

Ghias (1010) 6:23-28

8 Q So when you wrote "proportional to the log
9 of the size of the dataset," you meant that to mean
10 the query dataset?

11 A Yes. Yes.

12 Q Not the -- not the database?

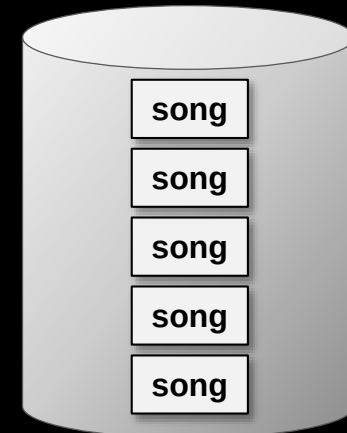
13 A That's -- that's correct. There's no
14 database here. It's a problem of matching a query
15 to a single song.

Moulin (2006-Part 2) 103:8-15

query



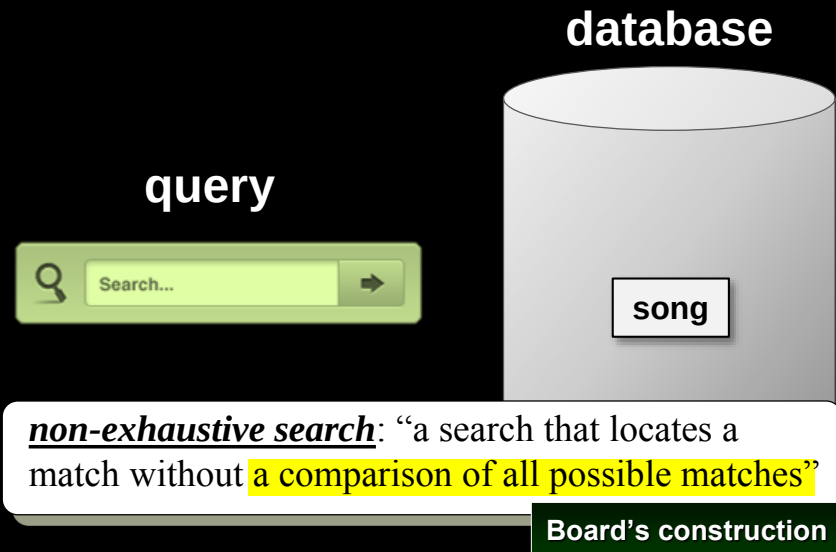
database



Petitioner's new Reply theory:

of the full database—as non-exhaustive. Ghias discloses string matching algorithms that operate faster than a brute force search. Paper No. 1 at 9–10 (citing Ex. 1010, 6:23–28 ("Running times have ranged from $O(mn)$ for the brute force algorithm to" substantially faster.) (emphasis modified)). While these algorithms often compare a query to a single reference, Ghias further discloses that "[p]erformance may . . . be improved by packing all the songs into one file." Ex. 1010, 7:41–45 (emphasis added). In this case, the entire search—not just the

'988 Reply at 7



		Independent claim	Key missing elements
'237	{	1	sub-linear-time search
		5	sub-linear-time search
		9	approximate nearest neighbor
		13	approximate nearest neighbor
'988	{	15	non-exhaustive search search identifying a neighbor determining an action based on the identification
		1	non-exhaustive search neighbor search
'179 & '441	{	13	non-exhaustive search neighbor search
		25	non-exhaustive search neighbor search

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

Claims 9 and 13 of the '237 patent further require that the search locate an "approximate nearest neighbor." Ex. 1001 at Claims 9, 13. Ghias discloses that this search locates a neighbor by determining "a ranked list of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody."

'237 Petition at 42

Rank

- | | |
|---|--------------------|
| 1 | Song L (1 error) |
| 2 | Song M (5 errors) |
| 3 | Song A (17 errors) |
| 4 | Song X (22 errors) |
| 5 | Song P (62 errors) |
| 6 | Song D (95 errors) |

single most
approximate
matching melody

Song L (1 error)

Petitioner's new Reply theory:

Ghias' subsequent searches do not always consider, let alone identify, the closest match. Ghias describes a first search, which considers all reference works

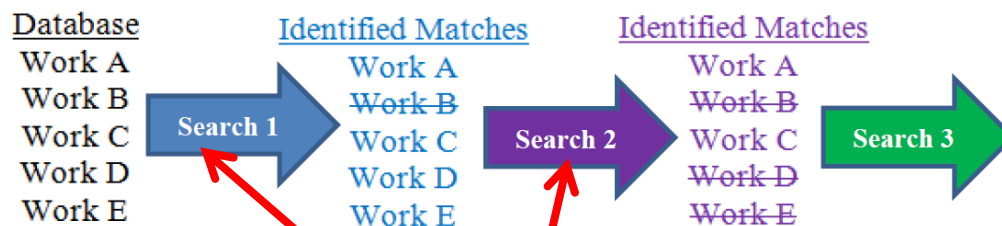
'237 Reply at 21

Claims 9 and 13 of the '237 patent further require that the search locate an "approximate nearest neighbor." Ex. 1001 at Claims 9, 13. Ghias discloses that this search locates a neighbor by determining "a ranked list" of approximately matching melodies, as illustrated at 26" or "the single most approximate matching melody."

'237 Petition at 42

Petitioner's new Reply theory:

Ghias' subsequent searches do not always consider, let alone identify, the closest match. Ghias describes a first search, which considers all reference works, followed by subsequent searches to refine the results, as shown below. Ex. 1010, 7:5–8. "[T]he same search algorithm is applied to [each] 'new query.'" Ex. 2005, ¶ 287. Thus, like the first search, each subsequent search can be further refined with a new query, which in turn will display a more refined "list consisting of songs just retrieved." Ex. 1010, 7:5–8.

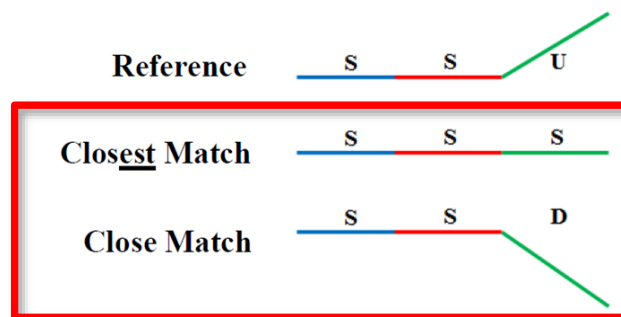


'237 Reply at 21

approximate nearest neighbor search: identifying a close match that is not necessarily the closest match

Board's construction

Petitioner's new Reply theory:



'237 Reply at 23

Dr. Karpyis confirmed that Ghias cannot distinguish this quality difference, and will consider both strings "equal quality melodies." See Ex. 1020, 164:7–21,

Petitioner's new Reply theory:



'237 Reply at 23

Dr. Karpyis confirmed that Ghias cannot distinguish this quality difference, and will consider both strings "equal quality melodies." See Ex. 1020, 164:7–21,

Rank

- 1 Song L (1 error)
- 2 Song M (5 errors)
- 3 Song A (17 errors)
- 4 Song X (22 errors)
- 5 Song P (62 errors)
- 6 Song D (95 errors)

single most
approximate
matching melody

Song L (1 error)

	Independent claim	Key missing element(s)
'237	1	sub-linear-time search
	5	sub-linear-time search
	9	approximate nearest neighbor
	13	approximate nearest neighbor
'988	15	non-exhaustive search search identifying a neighbor determining an action based on the identification
'179 & '441	1	non-exhaustive search neighbor search
	13	non-exhaustive search neighbor search
	25	non-exhaustive search neighbor search

**'179
&
'441**

Independent claim	Key missing elements
1	neighbor search non-exhaustive search
13	neighbor search non-exhaustive search
25	neighbor search non-exhaustive search

'179
&
'441

Independent claim	Key missing elements
1	neighbor search non-exhaustive search
13	neighbor search non-exhaustive search
25	neighbor search non-exhaustive search

neighbor search: “identifying a close, but not necessarily exact or closest, match”

Board’s construction

- ❑ a search that can only identify exact matches is not a neighbor search

92. Feature 1: 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 match 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.

Karypis ‘179 Decl. (2006) ¶92

neighbor search: “identifying a close, but not necessarily exact or closest, match”

Board’s construction

- ❑ a search that can only identify exact matches is not a neighbor search

somewhat different sets of extracted features. Ex. 1001 at 8:47-9:2. Thus, rather than design systems that searched only for exact matches, persons skilled in the art typically designed systems that identified close, but not necessarily exact, matches. *Id.* Persons skilled in the art understood that the term “identifying a neighbor” referred to such searches.

Moulin ‘179 Decl. (1004) ¶45

Google's desired claim:

Most importantly, the language of the independent claims at issue explicitly recites "identifying . . . the first electronic work . . . using a . . . neighbor search." Ex.

'179 Reply at 8

Google's desired claim:

result: (c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work
by: by using a non-exhaustive neighbor search;

claim 13

Most importantly, the language of the independent claims at issue explicitly recites "identifying . . . the first electronic work . . . using a . . . neighbor search." Ex.

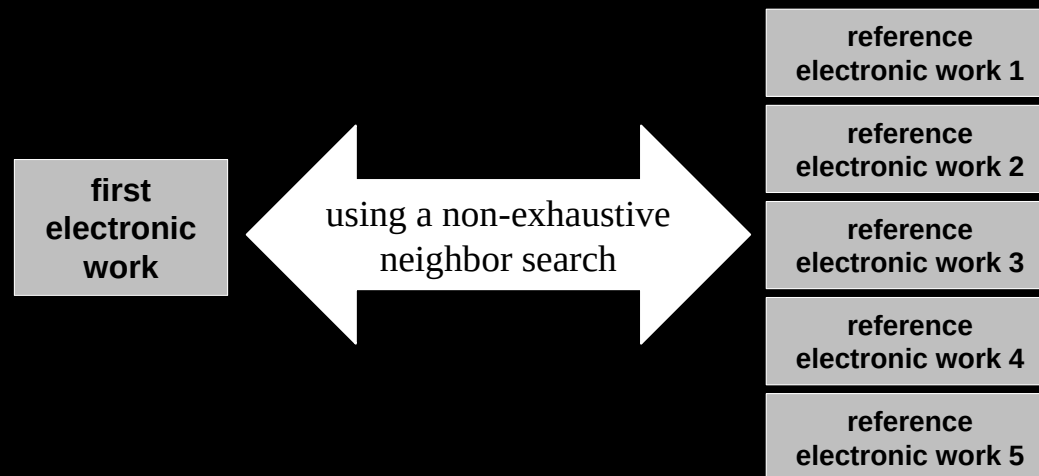
'179 Reply at 8

Google's desired claim:

result: (c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work

by: by using a non-exhaustive neighbor search;

claim 13



Actual claim:

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

claim 13

Google's desired claim:

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

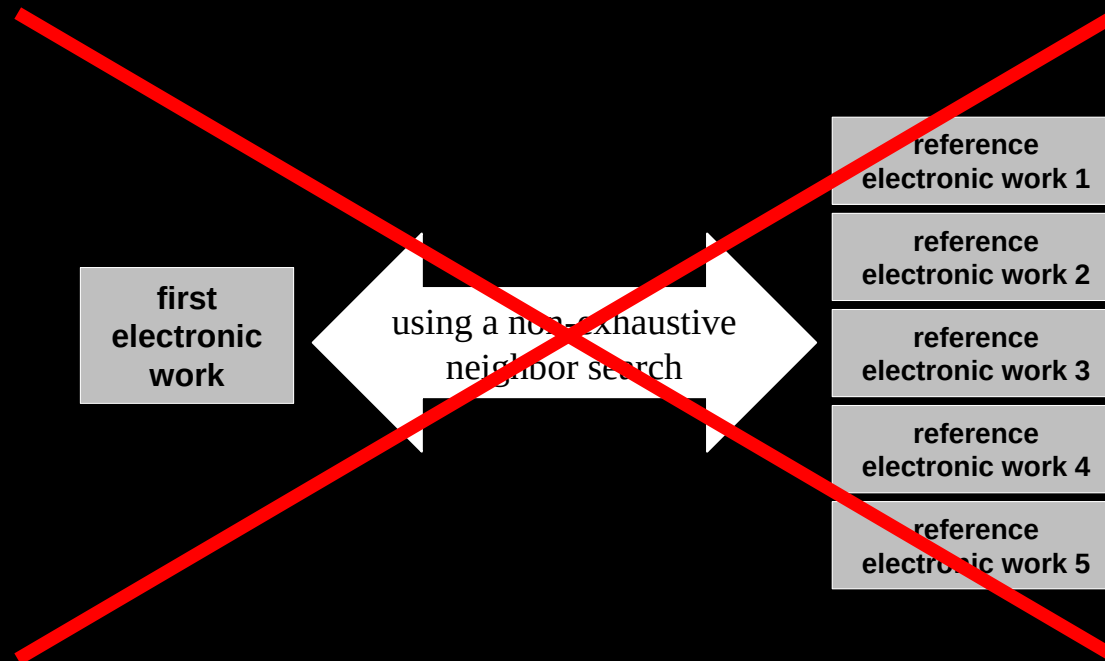
claim 13

Actual claim:

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

by:

claim 13

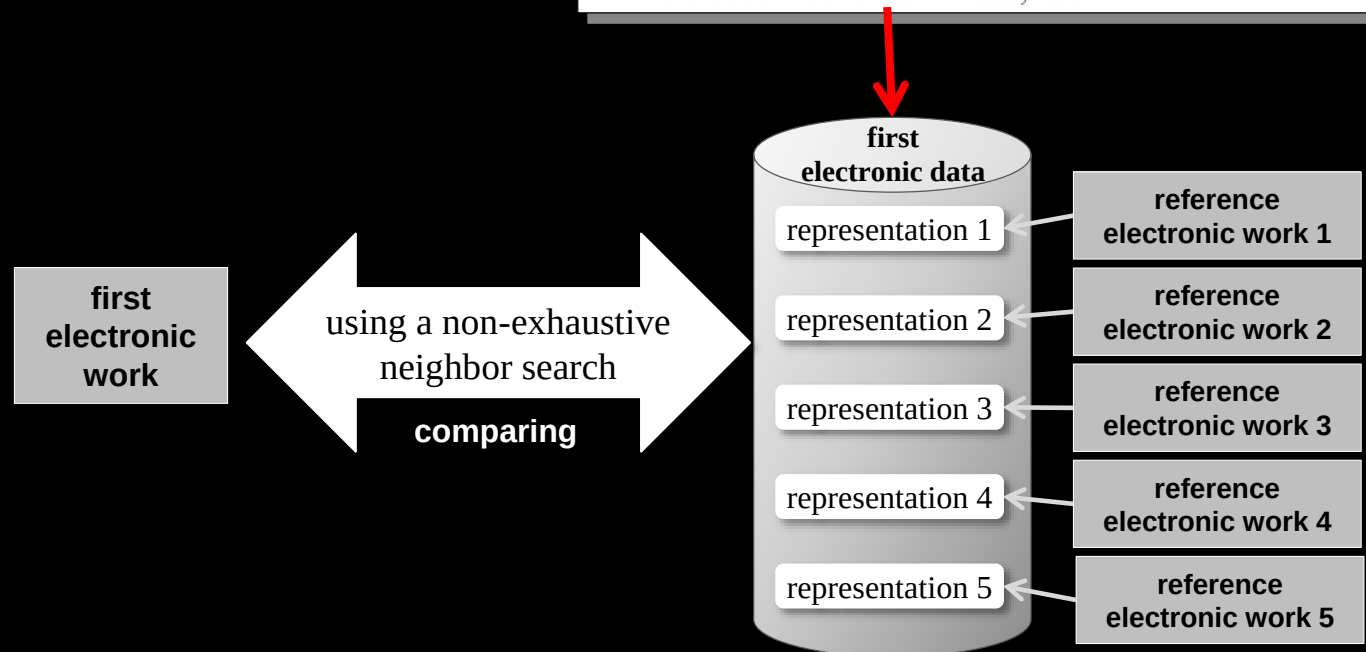


Actual claim:

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

claim 13

(a) maintaining, by a computer system, one or more databases comprising:
(1) first electronic data comprising a first digitally created compact electronic representation of one or more reference electronic works; and

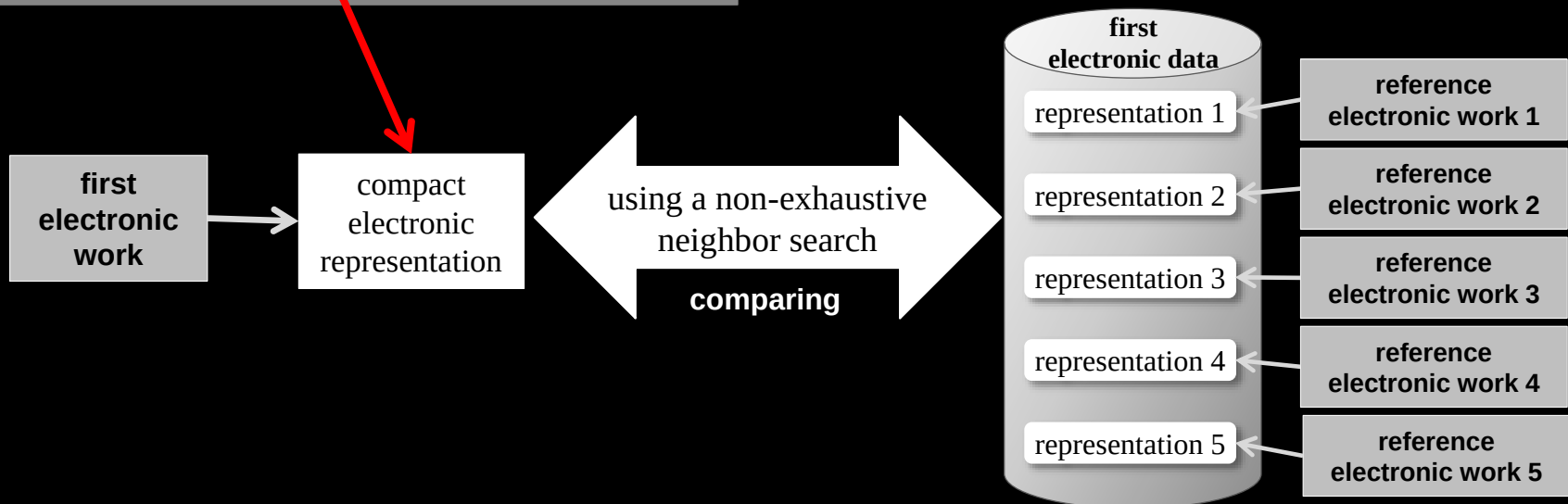


Actual claim:

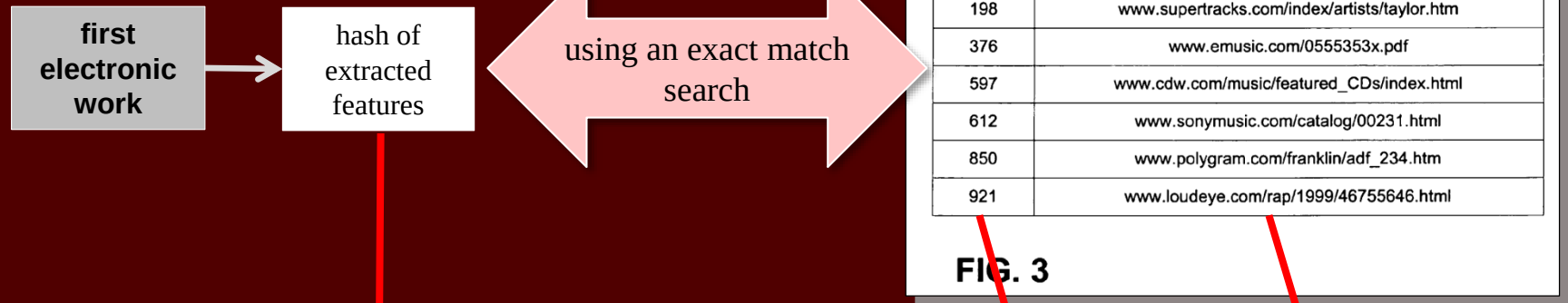
(c) identifying, by the computer system, a matching reference electronic work that matches the first electronic work by comparing the first electronic data with the second digitally created compact electronic representation using a non-exhaustive neighbor search;

claim 13

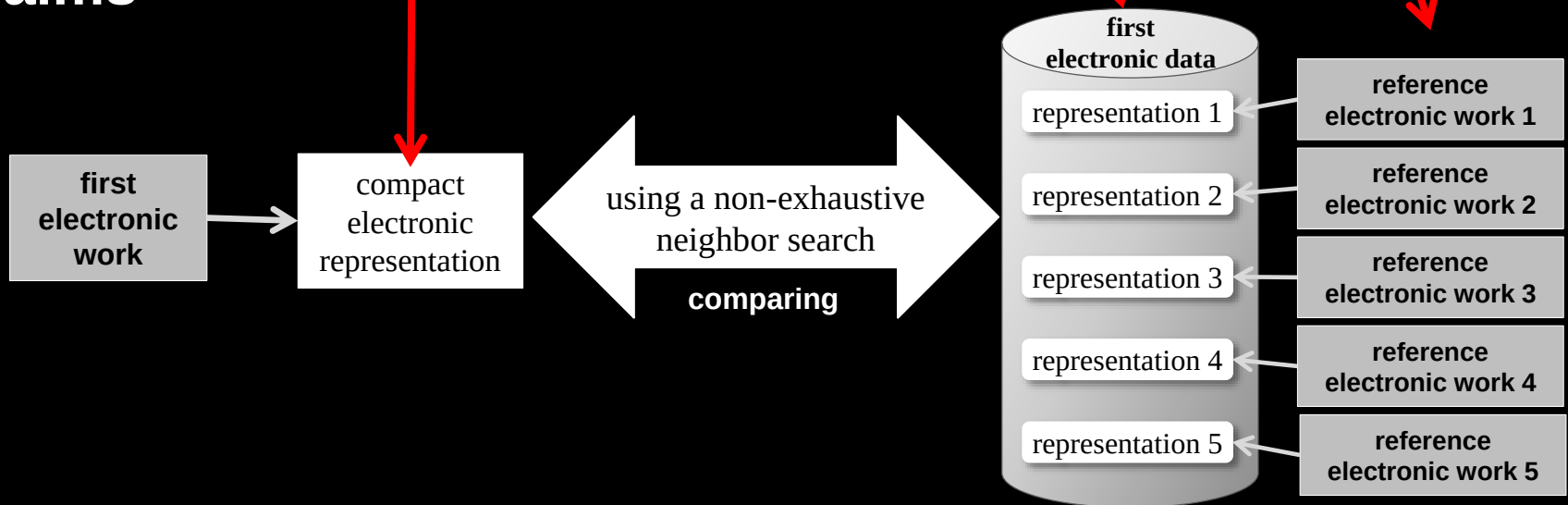
(b) obtaining, by the computer system, a second digitally created compact electronic representation of a first electronic work;



Conwell



Claims



Conwell

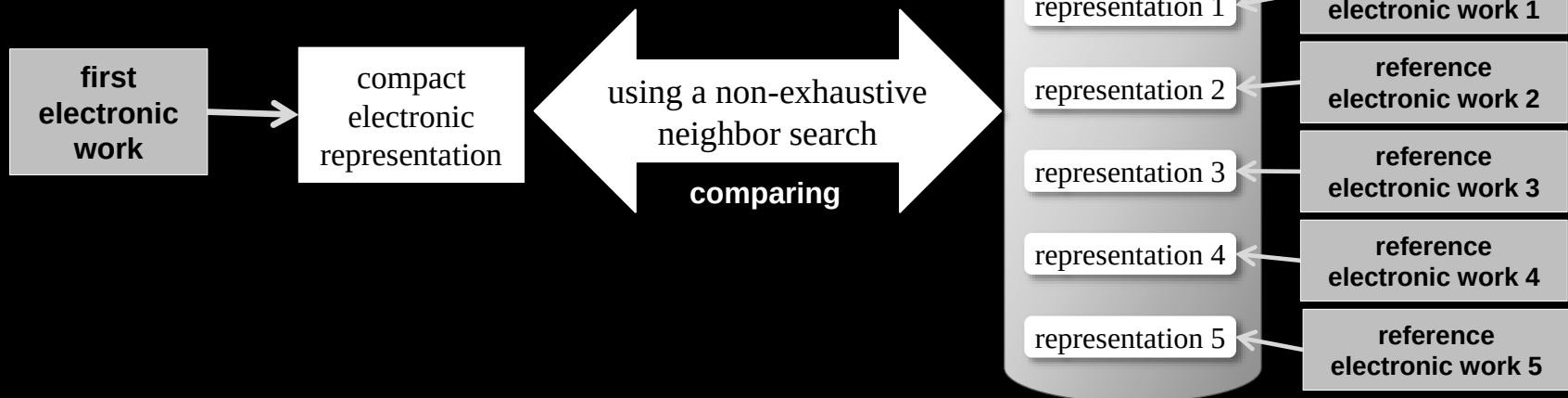


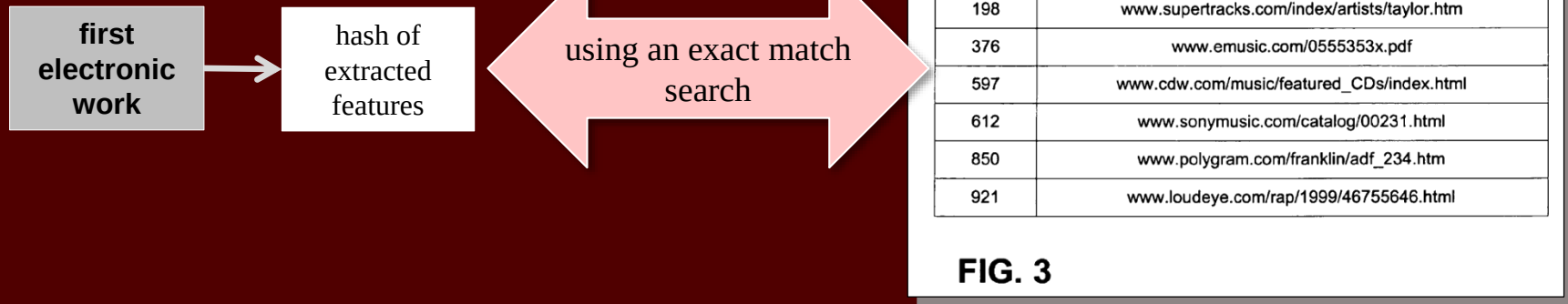
hash values	reference works
034	www.sonymusic.com/catalog/05634.html
112	www.sonymusic.com/catalog/00014.html
198	www.supertracks.com/index/artists/taylor.htm
376	www.emusic.com/0555353x.pdf
597	www.cdw.com/music/featured_CDs/index.html
612	www.sonymusic.com/catalog/00231.html
850	www.polygram.com/franklin/adf_234.htm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

FIG. 3

13 Q If -- in Conwell, when it does a
 14 comparison of the extracted features of the first
 15 electronic work with the first electronic data, it's
 16 comparing a hash value with a set of hash values;
 17 right?
 18 A Yes.

Moulin 264:13-25





13 Q If -- in Conwell, when it does a
 14 comparison of the extracted features of the first
 15 electronic work with the first electronic data, it's
 16 comparing a hash value with a set of hash values;
 17 right?

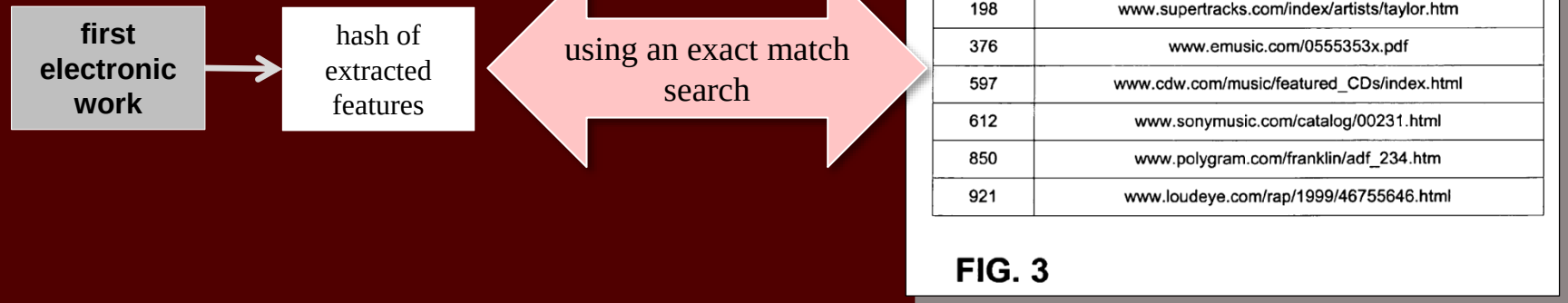
18 A Yes.

19 Q That comparison is always going to produce
 20 either an exact match or no match; right?

21 A Yes.

22 Q That -- that comparison will never return
 23 a suggestion that, "Here's something that doesn't
 24 quite match, but it's close"?

25 A That's correct.



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

Moulin (2006-Part 3) 266:21-24

'179
&
'441

Independent claim	Key missing elements
1	neighbor search non-exhaustive search
13	neighbor search non-exhaustive search
25	neighbor search non-exhaustive search

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

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

‘179 Petition at 24

034	
112	
198	
376	
597	
612	
850	
921	

FIG. 3

Karypis Decl. ¶¶365-385

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

**Karypis Decl.
(A2005) ¶ 377**

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

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

‘179 Petition at 24

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

FIG. 3

Karypis Decl. ¶¶365-385

Conwell, Figure 3.

367. As I explained above, such a search would be exhaustive rather than the claimed non-exhaustive because it systematically checks whether each potential match matches the work to be identified until a match is found.

Karypis Decl. (A2005) ¶ 367

Petitioner's responsive argument:

Cox '179

are not "exhaustive searches." Rather, the patent explains that database lookups commonly employed "binary search" (Ex. 1001, 21:37–40), which Dr. Karypis agrees is "non-exhaustive." Ex. 1020, 186:6–8. Consequently, the patent confirms

'179 Reply at 10

the test is not what is common but rather what is actually disclosed

proper claim construction. *Id.* at 1344. A claim limitation is inherent in the prior art if it is necessarily present in the prior art, not merely probably or possibly present. *Rosco*

Akamai Technologies, Inc. v. Cable & Wireless Internet Servs., Inc.,
344 F.3d 1186, 1192 (Fed. Cir. 2003)

Petitioner's New Reply Theory:

Accordingly, when a "genus is so limited that a person of ordinary skill in the art can, at once envisage each member of this limited class. . . . a reference describing the genus anticipates every species within the genus." *Id.* at 1337–38 (citation

'179 Reply at 11

Petitioner's New Reply Theory:

Here, not only would a skilled artisan have been able to envisage the constituent members of the database query "genus" (non-exhaustive and exhaustive), but one would have envisaged a "non-exhaustive" search as the *predominant* form of "database query." See Ex. 2005, ¶ 371. Dr. Karypis admits

'179 Reply at 11

Petitioner's New Reply Theory:

Here, not only would a skilled artisan have been able to envisage the constituent members of the database query "genus" (non-exhaustive and exhaustive), but one would have envisaged a "non-exhaustive" search as the *predominant* form of "database query." See Ex. 2005, ¶ 371. Dr. Karypis admits

'179 Reply at 11

Examples in '179:

- ☐ excluded middle vantage point forest for nearest search ('179, 9:17-18)
- ☐ sub-linear time search ('179, 4:34-35)
- ☐ approximate nearest neighbor search ('179, 4:36)
- ☐ binary search ('179, 8:53; 9:3-12)
- ☐ nearest neighbor search ('179, 9:7)
- ☐ kd-trees ('179, 9:14)
- ☐ vantage point trees ('179, 9:14)
- ☐ linear search ('179, 21:37-42)
- ☐ clustering ('179, 9:13)
- ☐ fixed radius search ('179, 22:19)
- ☐ classical textual search ('179, 23:6)

Petitioner's New Reply Theory:

Here, not only would a skilled artisan have been able to envisage the constituent members of the database query "genus" (non-exhaustive and exhaustive), but one would have envisaged a "non-exhaustive" search as the *predominant* form of "database query." See Ex. 2005, ¶ 371. Dr. Karypis admits

'179 Reply at 11

Examples in other evidence in the Record:

- ❑ Boyer-Moore (1012 – Iwamura, 9:52-53)
- ❑ neighbor searching (Paper 1 – Petition at 3, Paper 4 – Moulin at ¶12)
- ❑ lookup in a hash table (Paper 1 – Petition at 24)
- ❑ pattern matching (1015, 1:55, 2:8, 2:12, 5: 64-65, 6:7, 6:8, 6:43, 10:38; 10:58, 11:36, 12:26)
- ❑ approximate string matching (1010 – Ghias, 6:23-35)
- ❑ similarity search (1018 – “Similarity Search in High Dimensions via Hashing,” Abstract)
- ❑ priority search (1006 – Arya at 4)
- ❑ closest-point queries (1006 – Arya at 1)
- ❑ Peak search (1012 – Iwamura, 9:6)
- ❑ K-nearest neighbor (k-nn) query (1007 – “Efficient Similarity Search in Digital Libraries,” 2)
- ❑ Sequential search (1020 – Karypis Deposition Transcript, 70:3)

'179
&
'441

Independent claim	Key missing elements
1	neighbor search non-exhaustive search
13	neighbor search non-exhaustive search
25	neighbor search non-exhaustive search

11 Is it the case that for a search to be
12 sublinear as it's used in the '237 patent, it's not
13 enough for it to have execution time that is
14 sublinear in relationship to the size of the
15 pattern; it must also be sublinear in relationship
16 to the size of the database?

17 A When I read "sublinear" in, say, Claim 5
18 of the patent, as we just did, I understand
19 sublinear to mean in relation with the size of the
20 database. It does not say anything about in
21 relation with the size of the query.

Moulin (A2006-Part 1) 26:11-21

size of the query



sub-linear time search: “a search whose execution time scales with a less than linear relationship to the size of the data set to be searched”

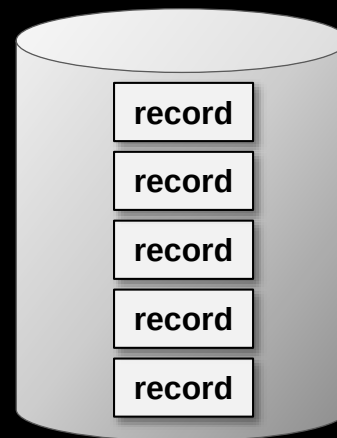
Board's construction

16 Q Well, as you understood the term
17 "sublinear" as it's used in the patent, it's about
18 the execution time as we increase the size of the
19 database; is that right?

20 A In the patent, yes.

Moulin (A2006-Part 2) 103:16-22

size of the data set



nonexhaustive search: “a search that locates a match without a comparison of all possible matches”

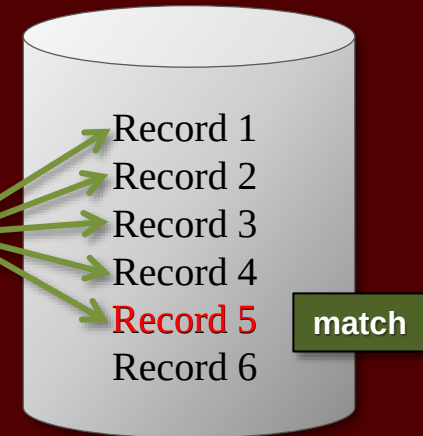
Board's construction

exhaustive search

dimensional) feature space. In previous work, it was not uncommon to perform a linear search of all N entries, perhaps halting the search when the first match is found. On average,

'237 8:59-61

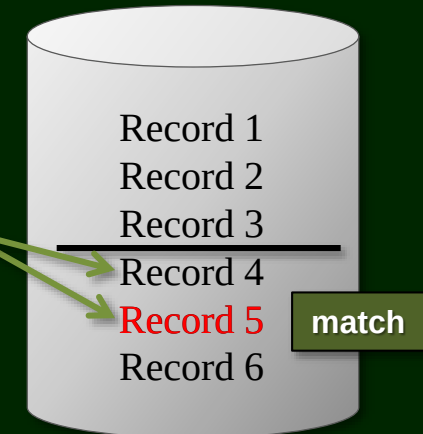
work to be identified



non-exhaustive search

Intelligent algorithm narrows the records in the database to be searched

work to be identified



non-exhaustive search

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board’s construction

non-exhaustive search: “a search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches.”

Petitioner’s construction

- ✓ ‘237 specification
- ✓ Objective evidence
- ✓ Petitioner’s Declarant
- ✓ Patent Owner’s Declarant

within each possible match. By contrast, Petitioner provides no other support for its construction other than its declarant, Dr. Pierre Moulin. See

‘237 Decision at 6

Claim Element	Recited In Claims
"non-exhaustive search"	25, 29–30
"approximate nearest neighbor search"	9, 11–12, 13, 15–16, 23–24, 33, 37–38
"sublinear search"	1, 3, 5, 7–8, 21–22, 33, 37–38

'237 Reply at 7

Patent Owner contends only that Iwamura does not disclose only (1) the "non-exhaustive search" element of independent claim 25; (2) the "approximate nearest neighbor search" element of independent claims 9, 13, and 33; and (3) the "sublinear search" element of independent claims 1, 5, and 33. Paper No. 17 at 11,

19, 25, 40. In fact, Iwamura discloses these elements and anticipate

'237 Reply at 6

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

In instituting Ground 1, the Board did not specifically address whether Iwamura disclosed the neighbor or near neighbor properties of the claimed search. Decision at 11-12. Iwamura does not disclose such properties. Karypis Decl.

¶¶196-203.

'237 Response at 39

identify a neighbor: identifying a close, but not necessarily exact or closest, match

Board's construction

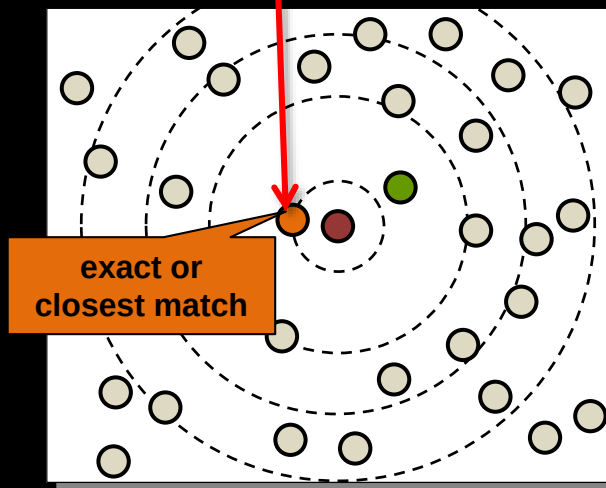
neighbor or near neighbor: a close, but not necessarily exact or closest, match

Board's construction

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.

'237 Petition at 10-11



identify a neighbor: identifying a close, but not necessarily exact or closest, match

Board's construction

neighbor or near neighbor: a close, but not necessarily exact or closest, match

Board's construction

11 Q If we implement that feature of Iwamura,
12 is it the case that by skipping a repeated pattern,
13 it's not going to stop us from producing the best
14 match?

15 A Again, if truly -- assuming it's truly a
16 repeated pattern, and, again, assuming it's weighted
17 correctly, indeed it will not affect the ability to
18 find the best match.

Moulin (A2006-Part 4) 318:11-18

14 Is it the case that if we do the Iwamura
15 search using the peaks as our basis, and we set up
16 our database such that the unimportant peaks are
17 skipped, that we're still going to be identifying
18 the closest match when we produce our results?

19 A That would be assuming that no peaks have
20 been dropped and everything we discussed yesterday.
21 Dropping an unimportant part is not going to affect
22 the ability to find the best match.

Moulin (A2006-Part4) 317:14-22

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

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

'988, Claim 15

☐ **identify electronic work**

then

☐ **based on the identified work, determine an action**

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

Petitioner incorporates the above discussion of Ghias regarding Claim 1d.

'988 Petition at 14

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

Ghias uses the results of a search to **determine the potential matches** (2:50-52, 6:60-63, 7:4-5, 8:26-28, 8:61-63), and how many potential matches to display to a user (6:60-65 ("The number of matches that the database 14 should retrieve depends upon the error-tolerance"). "If the list [of potential matches] is too large" the system may determine that it should perform the additional action of **allowing the user to perform "a new query on a restricted search list consisting of songs just retrieved."** *Id* at 7:5-7.

'988 Petition at 12

ranked list of matches

Song 1
Song 3
Song 5

sub-linear time search: “a search whose execution time scales with a less than linear relationship to the size of the data set to be searched”

Board's construction

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 $O(n\log(m))$ ")), which, as explained above in Section V(D), are sublinear (Ex. 1001 at 8:54-63).

'237, Moulin Decl. (1004) ¶ 123

Several Algorithms have been developed that address the problem of approximate string matching. Running times have ranged from $O(mn)$ for the brute force algorithm to $O(kn)$ or $O(n\log(m))$, where “ O ” means “on the order of,” m is the number of pitch differences in the query, and n is the size of the string (song). See Ricardo Baeza-Yates and G. H.

Ghias (1010) 6:23-28

16 Q My question, sir, is did you read this to
17 indicate or suggest that these algorithms would --
18 could be used to perform a sublinear time search
19 with respect to the size of the dataset being
20 searched?

21 A No.

Moulin (A2006-Part 1) 90:16-21

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

Moulin (A2006-Part 1) 89:14-18

non-exhaustive search: “a search that locates a match without a comparison of all possible matches”

Board's construction

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

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

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

12 A Yes.

Moulin (A2006-Part 4) 336:3-12

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

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

10 Q Yes.

11 A No, it does not teach that.

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

14 A Yes.

Moulin (A2006-Part 4) 338:6-14