

Filed on Behalf of NETWORK-1 TECHNOLOGIES, INC.

By: Charles R. Macedo (Reg. No. 32,781)
Brian A. Comack (Reg. No. 45,343)
Amster, Rothstein & Ebenstein LLP
90 Park Avenue
New York, NY 10016
Telephone: (212) 336-8074
Facsimile: (212) 336-8001
cmacedo@arelaw.com
N1-Google-IPR@arelaw.com

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
Patent 8,656,441

**PATENT OWNER'S PRELIMINARY RESPONSE TO THE PETITION
FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,656,441 UNDER 35
U.S.C. § 313 PURSUANT TO 37 C.F.R. § 42.107**

Table of Contents

I.	Introduction.....	1
II.	Overview of the ‘441 Patent.....	3
III.	Claim Constructions.	5
	A. “non-exhaustive.”	5
	B. “neighbor search.”	9
IV.	Ground 1: Obvious over Levy (Ex. 1013) in view of Arya (Ex. 1006) fails.....	10
	A. Overview of Levy.....	10
	B. Overview of Arya.....	13
	C. Ground 1 fails because the references fail to teach claim elements and teach away from the proposed combination.	16
	1. Levy and Arya teach away from combining these references.	17
	2. The two asserted motivations to combine Levy and Arya presented in the Petition both fail.	21
V.	Ground 2: Anticipated by Conwell (Ex. 1009) fails.	24
	A. Overview of Conwell.	24
	B. Ground 2 fails because Conwell does not disclose a “non- exhaustive neighbor search.”	26
VI.	Ground 3: Anticipated by Wood (Ex. 1015) fails.	30
	A. Overview of Wood.	30
	B. Ground 3 fails because Wood does not disclose a “non- exhaustive neighbor search.”	31
VII.	Ground 4: Obvious over Ghias (Ex. 1010) in view of Philyaw (Ex. 1014) fails.	34
	A. Overview of Ghias.....	34
	B. Overview of Philyaw.....	35
	C. Ground 4 fails because the Petition relies on Ghias for a “non- exhaustive neighbor search” and Ghias does not disclose this claimed element.....	36

VIII. Conclusion.	40
------------------------	----

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Glaxo Group Ltd. v. Apotex, Inc.</i> , 376 F.3d 1339 (Fed. Cir. 2004)	13, 26
<i>In re Gordon</i> , 733 F.2d 900 (Fed. Cir. 1984)	17
<i>International Securities Exchange v. Chicago Board Options Exchange</i> , IPR2014-00097, Paper 12 (PTAB May 22, 2014)	16, 28, 31, 38
<i>McGinley v. Franklin Sports, Inc.</i> , 262 F.3d 1339 (Fed. Cir. 2001)	17
<i>Schering Corp. v. Geneva Pharms., Inc.</i> , 339 F.3d 1373 (Fed.Cir.2003)	13, 26
<i>Schumer v. Lab. Computer Sys., Inc.</i> , 308 F.3d 1304 (Fed. Cir. 2002)	8
<i>Smart Modular Technologies v. Netlist, Inc.</i> , IPR2014-01371, Paper 12 (PTAB March 13, 2015)	8
Statutes	
35 U.S.C. § 316(e)	17
Other Authorities	
37 C.F.R. § 42.104(b)(4)	3

Exhibit List

EXHIBIT NO.	DESCRIPTION
Exhibit 2001	“Brute-force search”— http://en.wikipedia.org/wiki/Brute-force_search (3/19/2015)
Exhibit 2002	U.S. Patent 8,447,762 (Brendel)
Exhibit 2003	U.S. Patent 7,167,984 (Graveman)

I. Introduction.

The references asserted in the Petition fall into two general categories:

Category 1 consists of patent references that do not teach a “non-exhaustive neighbor search” to identify matches as required by each challenged claim. The patent references are Levy (Ex. 1013), Conwell (Ex. 1009), Ghias (Ex. 1010), Wood (Ex. 1015), and Philyaw (Ex. 1014). Rather than teaching to identify a work using the claimed “non-exhaustive neighbor search,” these references teach using a different type of search to identify matches:

- Levy teaches using exact match comparisons rather than neighbor searches to identify a match using an extracted hashed fingerprint (Levy, 9:42-61);
- Conwell teaches using the same exact match comparisons rather than neighbor searches to identify a match using extracted hashed features (Conwell, Abstract);
- Wood teaches an exhaustive search that compares the “specimen” with “each entry in the pattern library” (Wood, 6:8-23);
- Ghias teaches an exhaustive search that “compare[s] all the songs” in the reference database (Ghias, 5:66-6:2); and
- Philyaw does not disclose any specific searching algorithm.

Category 2 consists of an academic article—Arya (Ex. 1006)—that

identifies a neighbor search but does not disclose other claim elements. Because Arya address neighbor search algorithms rather than exact match comparisons, Arya teaches search methods that (a) produce a meaningless “match” if the extracted features are hashed (as taught by Levy) before conducting the disclosed neighbor search, and (b) are less efficient than exact match searches.

Each ground in the Petition fails.

Ground 1 (obvious over Levy in view of Arya) fails because the proposed combination, based on flawed premises, would not work, and the references teach away from such a combination rather than provide any motivation to combine.

Grounds 2 and 3 (anticipated by Conwell and Wood) fail because the references are missing key claimed elements—*e.g.*, a “non-exhaustive neighbor search.” And while the Petition proposes constructions, it fails to map its constructions onto these references. For example, Petitioner construes “neighbor search” as “identifying a close, but not necessarily exact match.” Pet. 6. But after asserting its construction, Petitioner’s construction appears nowhere in its Petition. The critical word in its constructions—“close”—never appears again.

Ground 4 (obvious over Ghias in view of Philyaw) fails because the proposed combination is missing key claimed elements—*e.g.*, “non-exhaustive neighbor search.”

The Petition fails to carry its burden under 37 C.F.R. § 42.104(b)(4) on each challenged independent claim (1, 13, and 25) and thus fails to carry its burden on any of the challenged dependent claims (which all depend on the challenged independent claims). The Petition should be denied in its entirety.

II. Overview of the ‘441 Patent.

The ‘441 Patent involves a search that compares a given item to a reference database of potential matches. This Preliminary Response uses the term “work” to mean the item (*e.g.*, a digital audio file) to be identified using the search. *See* ‘441, 6:49-54. This Response uses “record” to mean one of the units in the reference database that the work may be compared to. *See* ‘441, 6:15-18. And the Response use “database” or “library” to mean the collection of all records. *See* ‘441, 6:24-30.

The independent claims of the ‘441 Patent (claims 1, 13, and 25) include the following five elements for identifying a work and performing a corresponding action:

- [1] a database with (a) first data related to records, and (b) second data related to action information corresponding to the records;
- [2] extracting features from a work;
- [3] identifying the work based on the extracted features by performing “a

searching for a matching feature vector ('441, 6:44-48);

- “work-associated information lookup operation(s)” (160) that retrieve associated information, such as an action ('441, 6:49-51); and
- “action initiation operation(s)” (170) that performs some action based on the associated information ('441, 6:53-54).

The invention claimed in the '441 Patent includes two distinguishing features relevant to this Preliminary Response: (1) the “identifying” must be performed using a “non-exhaustive neighbor search;” and (2) the system must determine or associate an “action” based on the identified work. It is not sufficient to simply to identify a match. Rather, “an action” associated with the match must be “determined” or “associated.” '441, claims 1, 13, and 25.

III. Claim Constructions.

A. “non-exhaustive.”

A “non-exhaustive” search is a search using an algorithm designed to locate a match without requiring the work to be compared to all records in the database. A “non-exhaustive” search uses an intelligent algorithm to narrow the database to only a subset of potential matches. *See* Ex. 1004 ¶12 (“algorithms that increased efficiency by intelligently searching only a subset of potential matches (*i.e.*, ‘non-exhaustive’ algorithms)”). For example, if there are 100 records in a database, a

non-exhaustive search could use an intelligent algorithm to exclude 75 records from the search such that only 25 would be searched during the comparison process. As the specification of the '441 Patent observes, these non-exhaustive “forms of matching include those based on clustering, kd-trees, vantage point trees and excluded middle vantage point forests” that do not systematically compare the work to each record. '441, 8:65-67.

A “non-exhaustive” search can be contrasted with an “exhaustive” search. An exhaustive search systematically checks whether each record matches the work. An exhaustive search is “a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.” Ex. 2001.¹ If there are 100 records in the database, an “exhaustive” search does not narrow the potential matches but instead systematically compares the work with each record to determine if there is a match. Systematically comparing the work to be identified with each potential match rather than using intelligence to narrow the search candidates is also referred to as using “brute force.” Ex. 1004 ¶41 (“a brute force search conducts a comparison of every item in a search database”); Ex. 2001.

The Petition asserts that a “non-exhaustive search” should be construed as “a

¹ Emphasis added throughout this Response, unless otherwise noted.

search that locates a match without conducting a brute force comparison of all possible matches, and all data within all possible matches.” Pet. 5-6. The “all data” clause (underlined above) in Petitioner’s proposed construction would improperly include as a “non-exhaustive” search any search that did not compare “all data” in each record, even if the search was 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, according to Petitioner, the search is not “exhaustive” even though every record is compared.

To support its proposed construction, Petitioner relies solely on the conclusory assertions in its declaration (Pet. 5-6), and fails to present any objective evidence (*e.g.*, cites to the ‘441 Patent specification and prosecution history, dictionary definitions, or treatises). The “all data” clause is improper because it is:

- inconsistent with how the non-exhaustive search concept is used in the ‘441 Patent 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*

'441, 21:5-56; 8:-9:61); and

- not part of the ordinary meaning of “non-exhaustive search” (*see* Ex. 2001).

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.” Ex. 1004

¶40. The declarant provides no analysis or support for this conclusory assertion that is, therefore, insufficient to meet Petitioner's burden. *Smart Modular Technologies v. Netlist, Inc.*, IPR2014-01371 Paper 12 at 16 (March 13, 2015) (“testimony is insufficient if it is merely conclusory” (quoting *Schumer v. Lab. Computer Sys., Inc.*, 308 F.3d 1304, 1315 (Fed. Cir. 2002))).

Moreover, objective sources confirm 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 exhaustive search, 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 checking whether each candidate satisfies the problem's statement.

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

In fact, Petitioner's own declarant twice confirmed that a “non-exhaustive”

search searches a subset of “potential matches,” not a subset of “all data within all potential matches”:

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).

Ex. 1004 (Petitioner's declaration) ¶12.

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.

Ex. 1004 (Petitioner's declaration) ¶40.

As demonstrated below, the Petition fails under either the proper construction of “non-exhaustive” or under Petitioner's improper construction that includes the “all data” clause.

B. “neighbor search.”

Petitioner asserts that “neighbor search” should be construed to mean “identifying a close, but not necessarily exact, match.” Pet. 6. Network-1 defines “identifying a neighbor” as (i) identifying a close, but not necessarily an exact or the closest, match of a feature, (ii) where “close” means a distance or difference that falls within a defined threshold, and (iii) where “feature” means a feature

vector, compact electronic representation, or set of extracted features.

The parties effectively agree on the first part of Network-1's construction—"identifying a close, but not necessarily an exact or the closest, match." The second part clarifies what constitutes a "close" match as required by the '441 specification.² The third part clarifies what is being compared in the various claims of the '441 and related patents. As demonstrated below, the Petition fails under either construction of "neighbor search."

IV. Ground 1: Obvious over Levy (Ex. 1013) in view of Arya (Ex. 1006) fails.

A. Overview of Levy.

Levy identifies three ways to link "audio and other multimedia data objects with metadata and actions via a communication network" (Levy, 1:42-44): (1) using a "file header embedder" (Levy, 7:64-8:19); (2) using a "steganographic embedder" (*e.g.*, digital watermark) (Levy, 8:20-9:28); or (3) using "dynamic

² The '441 specification describes "neighbor" searches as determining a match (or lack of match) based on some threshold. *See e.g.*, '441, 6:46-48 ("If a match, or a match within a predetermined threshold is determined, then the associated work identifier is read."); *id.* 21:52-54 ("if the distance between the query and the nearest neighbor exceeds a threshold, then they are considered not to match.").

derivation from content” (Levy, 9:40-10:17). Only the third way is relevant to the Petition and is described below.

Levy teaches a “dynamic derivation” system that: (a) extracts features from an audio signal by hashing the audio signal to obtain a unique fingerprint identifier; and (b) searches for an exact match of the unique fingerprint identifier in the library:

the decoder computes a fingerprint of the audio signal based on a specified fingerprinting algorithm. The fingerprint is a number derived from a digital audio signal that serves as a statistically unique identifier of that signal. ... The hash algorithm may be applied to a selected portion of a music file (*e.g.*, the first 10 seconds) to create a fingerprint.

Levy, 9:46-55.

Levy teaches that a “decoder computes a fingerprint of the audio signal based on a specified fingerprinting algorithm” and states that “[o]ne component of the fingerprint algorithm is a hash algorithm.... Examples of hashing algorithms include MD5, MD2, SHA, SHA1.” Levy, 9:42-61. The characteristic of the hash algorithms disclosed in Levy (*e.g.*, MD5, MD2, SHA, and SHA1) is that they produce a unique output (a “unique identifier”—Levy, 9:46-55) that appears to be a random number:

Hash engine 44 may use a cryptographic hash function that produces a

seemingly random result compared with its input. Even a small change to the input value produces a large change in the hash output value when a cryptographic hash function is used.

Ex. 2002, 7:4-8. With respect to the extracted features using a hash algorithm, “even a single bit difference results in an entirely different hash value.” Ex. 2003, 2:1-10.

Assume for example that digital audio file “A” was hashed to a 16 digit number: 7630013485915355. Assume that one bit in file “A” was changed to produce file A₁. The hash of A₁ might look like this: 4421673894334691. As a result, a useful comparison of hashed files must look for an exact match. If the hash numbers of two files happen to be close (perhaps varying by only a single digit), that is simply a random coincidence, and provides no indication that the underlying audio files are close (one could be Bach and the other Led Zeppelin or even random noise). Searches based on such hashing algorithms, like the ones disclosed in Levy, do not include any concept of closeness—“close, but not necessarily an exact or the closest, match”—as required by the construction of a “neighbor search.” Pet. 6; Ex. 2002, 2:1-10.

Accordingly, when implementing the hash algorithms taught in Levy to identify a match, comparing the hashed extracted features from a work and a record in the library produces a binary result: either (1) an exact match; or (2) no

exact match. Whether the work and the record are close (similar) or distant (dissimilar) is not considered, is not relevant, and cannot even be determined using the hashed extracted fingerprints taught in Levy. Accordingly, these hashed fingerprints disclosed in Levy cannot be used to perform a “neighbor” search as required by each independent claim of the ‘441 Patent.

Levy does not disclose any method of extracting features (a fingerprint) of the audio signal to be compared to the potential matches in the reference database other than using these traditional hash algorithms. As a result, no other extraction technique is expressly or inherently disclosed—*i.e.*, “necessarily present”—in Levy. *Glaxo Group Ltd. v. Apotex, Inc.*, 376 F.3d 1339, 1348 (Fed. Cir. 2004) (quoting *Schering Corp. v. Geneva Pharms., Inc.*, 339 F.3d 1373, 1377 (Fed.Cir.2003)).

B. Overview of Arya.

Arya is an academic article that discloses neighbor searching algorithms that use extensive pre-processing of the data library for the algorithm to be effective. Specifically, Arya discloses an algorithm wherein “set S of n data points” is “preprocess[ed] . . . into a data structure, so that given any query point q . . . the k nearest neighbors of q can be computed.” Arya 1. Arya teaches that the disclosed algorithm can be used to find “k nearest neighbors” when accurate perfect match,

one-to-one comparison techniques (such as the hashed fingerprint comparison technique taught in Levy) cannot be used. Arya 1. Arya does not address exact match searches but instead provides an algorithm that can be used as a less accurate alternative when exact match searches are not available.

Moreover, Levy's exact match system is always more efficient (takes less time and processing power) than the neighbor search of Arya. This is because the number of comparison used in the neighbor search of Arya is always greater than the number of comparisons used in the exact match system of Levy. "If binary search was possible, then a database containing N vectors would require at most $\log(N)$ comparisons. '441, 8:55-56.

By contrast, Arya teaches that the nearest neighbor search disclosed in Arya requires $\underline{k}d\log(n)$ comparisons which is always greater than the $\log(n)$ because " k " (the number of neighbors to be identified) and " d " (the number of dimensions—data elements in the vector to be compared) are by definition greater than 1: "we show that given an integer $\underline{k} \geq 1$, $(1 + \epsilon)$ -approximations to the k nearest neighbor of q can be computed in additional $O(kd\log n)$ time." Arya 1. According, the binary exact match system of Levy is " kd " times more efficient than the neighbor search algorithm of Arya. (The '441 Patent similarly explains that a neighbor search is to be used when "binary search is not possible." '441, 8:55-60.)

Moreover, to improve the efficiency of its near-neighbor algorithm, Arya requires extensive pre-processing of the potential matches before the searches even take place. Arya states that by using very extensive and computationally complex “ideal” preprocessing of the data (that takes “ $n \log n$ time”), the search phase using a neighbor search could then approach the efficiency of an exact match search:

From the perspective of worst-case performance, an ideal solution would be to preprocess the points in $O(n \log n)$ time, into a data structure requiring $O(n)$ space so that queries can be answered in $O(\log n)$ time.

Arya 2.

If a data library is dynamic (as is the case with Levy), any performance efficiency associated with Arya's approach disappears as a result of the extensive preprocessing recalculations that must be performed on the constantly-changing data in the database library. This is because the pre-processing before any searching can even take place must be rebuilt when data elements are added to or removed from the data library. Accordingly, the efficiency of the search algorithm disclosed in Arya is missing, and the algorithm is not practical, for data libraries constituting dynamic data sets.

Arya does not teach any method of extracting data. Rather, Arya relies on the underlying existing system to provide the extracted features to be used by the

Arya algorithm.

C. Ground 1 fails because the references fail to teach claim elements and teach away from the proposed combination.

Ground 1 relies on Levy for all claimed elements except the “non-exhaustive neighbor search” limitation found in each independent claim. Pet. 11-20. The Petition does not rely on Levy for the “non-exhaustive neighbor search” limitation because, as explained above, Levy does not disclose a “non-exhaustive neighbor search.” *See* Section IV(A); Pet. 11-20. Instead, the Petition relies on Arya for this element which Petitioner asserts could be used in the Levy system.

With respect to Arya, however, the Petition (and corresponding declaration) fails to apply Petitioner's constructions (or any constructions) to Arya.

Accordingly, the Petition fails to satisfy its burden of demonstrating that Arya teaches a “non-exhaustive search identifying a neighbor” because it fails to “map the construed claim language to the teachings of the asserted prior art references.”

International Securities Exchange v. Chicago Board Options Exchange, IPR2014-00097, Paper 12 at 26 (May 22, 2014).

More importantly, as demonstrated below, Levy and Arya teach away from replacing the exact-match system disclosed in Levy with the neighbor search algorithm disclosed in Arya as proposed by Petitioner. As a result, the Petition fails to satisfy its burden of demonstrating a motivation to combine Levy and Arya

(35 U.S.C. § 316(e)), which teach away from such a combination.

First, this Preliminary Response demonstrates that Levy and Arya teach away from such a combination. Second, this Response demonstrates that the asserted motivations to combine Levy and Arya presented in the Petition fail.

1. Levy and Arya teach away from combining these references.

As explained above, each independent claim of the '441 Patent requires: [1] obtaining extracted features of an unknown work; and [2] identifying the unknown work using a “non-exhaustive neighbor search.” '441, claims 1, 13, and 25.

In Petitioner's proposed combination of ground 1, Petitioner relies on Levy for [1] extracting the features, and Arya for [2] identifying the work based on the features extracted using the Levy system. Pet. 11-20. Levy and Arya teach against combining the less efficient neighbor algorithm of Arya into the perfect-match searching system taught in Levy for two reasons.

First, incorporating the neighbor search algorithm of Arya into the exact-match system of Levy will create erroneous meaningless results. “If references taken in combination would produce a ‘seemingly inoperative device,’ we have held that such references teach away from the combination and thus cannot serve as predicates for a prima facie case of obviousness.” *McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1354 (Fed. Cir. 2001); *In re Gordon*, 733 F.2d 900, 902 (Fed.

Cir. 1984). Levy teaches a system that: (a) extracts features from an audio signal by hashing the audio signal to obtain a unique fingerprint identifier; and (b) searches for an exact match of the fingerprint in the library of potential matches by comparing the hashed fingerprint to hashes of the potential matches in the reference library:

the decoder computes a fingerprint of the audio signal based on a specified fingerprinting algorithm. The fingerprint is a number derived from a digital audio signal that serves as a statistically unique identifier of that signal. ... The hash algorithm may be applied to a selected portion of a music file (*e.g.*, the first 10 seconds) to create a fingerprint.

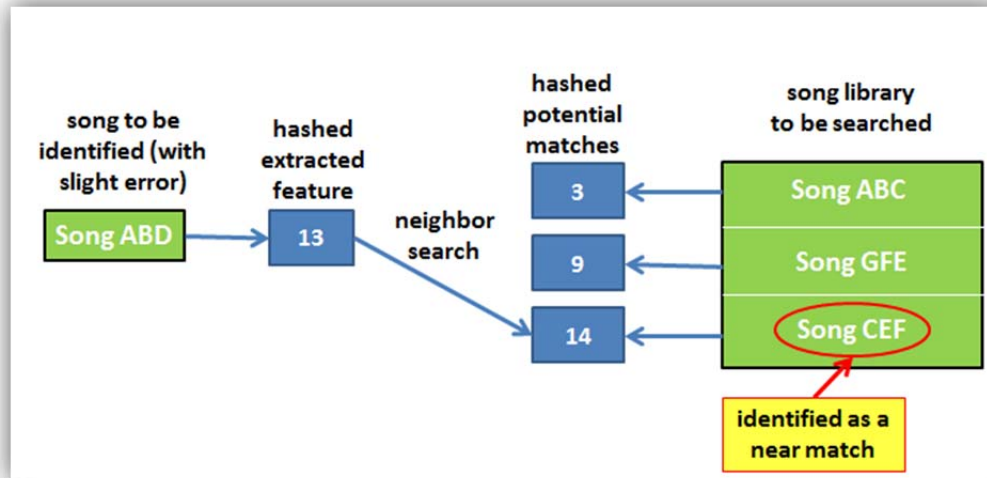
Levy, 9:46-55. Because the unique hash fingerprint of a work is compared to hash fingerprints of records in the library looking for an exact match, only perfect matches are relevant (as explained above, the concept of close does not count for hashed audio signals).

Trying to use the neighbor search algorithm of Arya in Levy system would identify “k nearest neighbors” (Arya, Abstract) of the hash of the work. Finding a neighbor of the hash fingerprint would not find an exact match, closest match, or even a close match to the work itself but instead would identify a random (and therefore erroneous) reference, as illustrated by the following simple example.

Assume a song to be identified is ABD (very close to ABC) and that the

three potential matches in the reference library are ABC, GFE, and CEF. Using the Levy system to extract an identifying fingerprint, ABD is hashed to produce a unique fingerprint, *e.g.*, “13.” The data of the songs in the library would also be extracted and hashed such that, for example, ABC is “3,” GFE is “9,” and CEF is “14.” There is no correlation between the pre-hashed songs and the hashed fingerprints because hashing eliminates any such relationship. This is because the output of standard hash functions, such as the MD5 and SHA-1 algorithms disclosed in Levy (Levy, 9:57-60), are dramatically changed even if one bit of input changed.

In this example, the exact match algorithm of Levy would correctly produce no match because, in fact, the library contains no match to ABD. The neighbor search algorithm of Arya, however, would erroneously identify CEF as a close or the closest match to ABD (13 is close to 14) when in fact ABC is actually the closest match.



Accordingly, identifying a close match of a fingerprint – extracted using a hash algorithm as taught in Levy – using the neighbor algorithm of Arya makes no sense and would produce erroneous matches, since close does not count when trying to match hashed data. Ex. 2002, 2:1-10. Because Levy uses an extraction technique that hashes the underlying audio signal to create the fingerprint, a search technique for anything other than an exact match produces nonsense.

Second, incorporating the neighbor algorithm of Arya into the exact-match system of Levy would not only produce nonsense, it will be less efficient in producing the nonsense. As explained above, the exact match binary search disclosed in Levy “requires at most $\log(N)$ comparisons” (where “N” is the number of records in the database). ‘441, 8:55-56. The approximate nearest neighbor search taught in Arya, by contrast is less efficient and takes “kd” times longer

(where “k” is the number of identified neighbors and “d” is the number of data points in the vector to be identified). Arya 1 (“we show that given an integer $k \geq 1$, $(1 + \epsilon)$ -approximations to the k nearest neighbor of q can be computed in additional $O(kd \log n)$ time.” (where “n” is the number of records in the database)) Therefore, efficiency teaches away from replacing the more efficient comparison technique of Levy with the less efficient neighbor algorithm of Arya.

Moreover, using the algorithm of Arya would require additional extensive and computationally complex preprocessing of all data in the database. Arya discloses that, even under theoretically “ideal” conditions, it would take “ $n \log n$ time” to just preprocess the data. Arya 2. Accordingly, Levy and Arya teach against combining the less efficient neighbor algorithm of Arya that would produce nonsense, into the more efficient exact-match searching system taught in Levy.

2. The two asserted motivations to combine Levy and Arya presented in the Petition both fail.

Petitioner's Motivation 1: Petition asserts that a person of ordinary skill in the art would have been motivated to use the neighbor search algorithm of Arya to search for a neighbor of the hashed extracted fingerprint of Levy because: (a) “disparities in the source of a fingerprinted work inevitably led to slight discrepancies in generated fingerprints;” (b) “persons skilled in the art understood that identifying a media work required not only locating exact fingerprint matches,

but also near fingerprint matches;” and (c) Arya teaches the concept of searching for an approximate nearest neighbor that could address these slight discrepancies.

Pet. 9. These assertions suffer from two fatal defects.

Defect 1: Levy specifically addresses the “slight discrepancies” identified by Petitioner. If a reference already adequately addresses a potential concern, one skilled in the art would not be motivated to search for another reference to address the already addressed concern. Levy specially addressed the concern associated with slight discrepancies in media works in two ways: by hashing (1) discreet samples, and (2) using attributes less sensitive to typical audio process, for example, a “low pass filtered version.” Levy, 9:55-59. Accordingly, one of ordinary skill in the art would not look to another reference (such as Arya) to address a problem that does not exist in Levy.

Defect 2: More importantly, the algorithm disclosed in Arya could not address any problem associated with slight discrepancies in the media work but instead would exacerbate any such problem. The premise underlying Petitioner’s argument is that when applying the neighbor algorithm to the hash extraction of Levy, if there are slight discrepancies in the underlying media work, a neighbor search could identify a close match if an exact match cannot be found. That premise is wrong.

Applying the neighbor search algorithm of Arya to the hash extraction of Levy would not identify a close match of the work but instead, would result in identifying a random record in the library (which is worse than identifying no match because inappropriate “actions” would be taken as a result of the false identification). This is because, as demonstrated above, when hashing data to extract a fingerprint, any difference in the source being hashed (*i.e.*, “slight discrepancies”) would result in a completely different hash (not a “close” hash) that would have no relationship to a hash of the data without the slight discrepancies. Once the work or a portion of the work is hashed, searching for a close match (as taught in Arya) rather than an exact match (as disclosed in Levy) is meaningless. Accordingly, searching for a “neighbor” of a hash fingerprint would result in identifying a random record in the library, rather than a neighbor to the work.

Petitioner's Motivation 2: Petitioner's declarant asserts that “a person skilled in the art would be motivated to employ the algorithm of Arya to implement the system of Levy in a more efficient manner than a brute force search would allow.” Ex. 1004 ¶61. As demonstrated above, replacing the more efficient exact match system of Levy with the less efficient neighbor search algorithm of Arya would make the system less efficient, not more efficient. Arya

does not state that the disclosed algorithm is more efficient than the exact match comparison taught in Levy. In fact, Arya confirms the exact opposite—that an exact matching search will always be more efficient than the disclosed neighbor search. Arya 2; *see* Section IV(C)(1). Accordingly, replacing the exact matching algorithm of Levy with neighbor algorithm taught in Arya would make the system less efficient rather than more efficient. The Petition provides no analysis or explanation as support for its assertion that inserting the algorithm of Arya into the system of Levy would make the system “more efficient” (Ex. 1004 ¶61) because it would do the opposite.

* * *

Ground 1 should not be instituted for the reasons set forth above.

V. Ground 2: Anticipated by Conwell (Ex. 1009) fails.

A. Overview of Conwell.

Conwell discloses associating media content, such as MP3 files, with identifiers and URLs. Conwell, Abstract.

Conwell (1) was assigned to the same entity as Levy (Digimarc Corporation), (2) includes a common inventor (Levy), (3) was filed the same month as Levy (May 2000), and specifically incorporates the related disclosure of Levy:

The subject matter of the present application is related to that disclosed in copending application Ser. No. 09/476,686, filed Dec. 30, 1999; 09/531,076, filed Mar. 18, 2000; Ser. No. 09/563,664, filed May 2, 2000; and 09/574,726, filed May 18, 2000 [Levy]. The disclosures of these applications are incorporated herein by reference.

Conwell, 1:7-1.

Conwell discloses two approaches to identifying a work: (1) assigning identifiers, or (2) implicitly generating identifiers derived from the data using a hashing algorithm. Conwell, Abstract. The implicit approach (rather than explicitly assigning identifiers) is the approach relevant to the Petition. Pet. 22-23.

Consistent with Levy (which is specifically incorporated by reference), Conwell relies on the same “hashing” algorithms to extract features from the work to be identified: “The identifiers can be assigned, or can be implicit (*e.g.*, derived from other data in the content object, as by hashing).” Conwell, Abstract; 1:65-67 (“some or all of the content data is processed by a hashing algorithm to yield a 128 bit identifier corresponding to that content.”)

Using such a hashing algorithm is only extraction approach taught in Conwell and Levy (incorporated by reference into Conwell). No other extraction technique is expressly or inherently disclosed—*i.e.*, “necessarily present.” Inherency “requires that the ‘missing characteristic is necessarily present,’” not

merely possible, or even probable. *Glaxo Group Ltd. v. Apotex, Inc.*, 376 F.3d 1339, 1348 (Fed. Cir. 2004) (quoting *Schering Corp. v. Geneva Pharms., Inc.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003)).

As explained above (with respect to Levy), the characteristic of such hash algorithms is that they are designed to produce a unique output for any given input. *See* Levy, 9:42-61; section IV(A) above. Accordingly, when implementing the hash algorithms taught in Conwell to identify a match, comparing the hashed extracted features from a work and a record in the library produces a binary result: either (1) an exact match; or (2) no exact match. Whether the work and the record are close (similar) or distant (dissimilar) is not considered, is not relevant, and cannot even be determined using a nearness comparison of the hashes extracted in Conwell.

B. Ground 2 fails because Conwell does not disclose a “non-exhaustive neighbor search.”

Each independent claim of the ‘441 Patent include a “non-exhaustive neighbor search” limitation. Claims 1, 13, and 25.

Conwell does not disclose the claimed “non-exhaustive neighbor search” but rather teaches an exact search. Similar to Levy, Conwell discloses only “hashing” algorithms to extract features to be compared. Conwell, Abstract. “The identifiers can be assigned, or can be implicit (*e.g.*, derived from other data in the content

object, as by hashing)." Conwell, Abstract.

As explained above, the characteristic of the hash algorithms disclosed in Conwell is that they are designed to produce a unique output for any given input. As a result, such hashing algorithms do not include any concept of closeness—"close, but not necessary exact"—as required even under Petitioner's construction of a "neighbor search." Pet. 6. Whether the work and record in the database are close (similar) or distant (dissimilar) is not (and cannot be) considered using the system taught in Conwell. Accordingly, these hashed identifiers disclosed in Conwell cannot be used to perform a "neighbor" search as required by each independent claim of the '441 Patent.

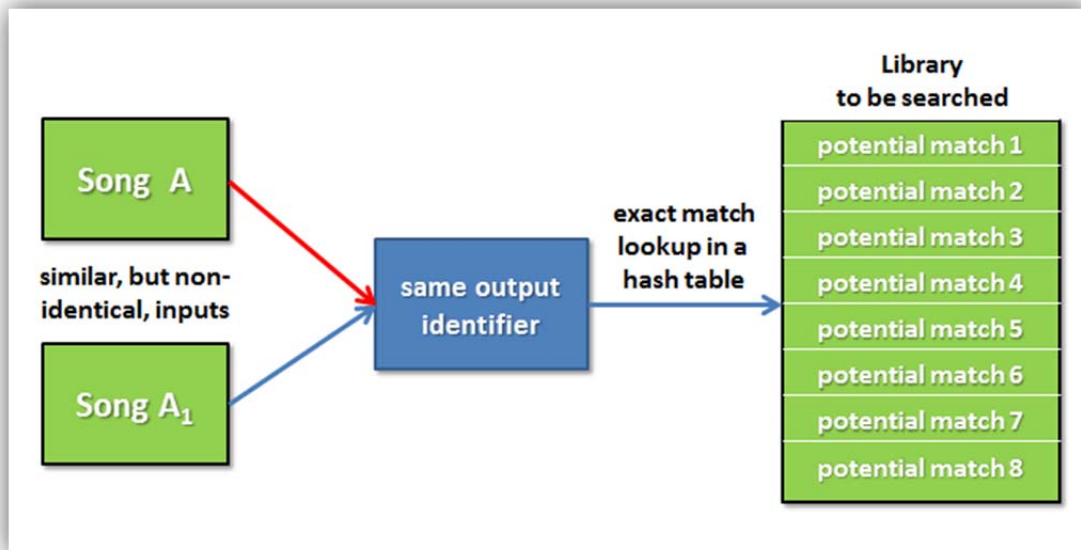
As support that Conwell discloses the claimed "non-exhaustive neighbor search," the Petition (and corresponding declaration) rely on the following assertion:

Conwell teaches table look-up using the identifier decoded from the electronic work. Ex. 1009 at 3:43-62, Figs. 1, 3. Because the hash algorithm generates the same output identifier for similar, but non-identical, inputs, the table look-up will return similar 'neighbor' results even when the input work is not identical to the reference work. Pet. 23 (citing Conwell, 4:64-5:3).

This statement in the Petition (and corresponding declaration) fails to apply

Petitioner's constructions (or any other constructions) to Conwell or explain how the disclosed hash algorithm explicitly or inherently results in a "neighbor" search. Accordingly, the Petition fails to satisfy its burden with respect to Conwell because it fails to "map the construed claim language to the teachings of the asserted prior art references." *International Securities*, IPR2014-00097, Paper 12 at 26.

Moreover, the Petition confirms that Conwell discloses an exact match search rather than the claimed "neighbor" search. According to the Petition, "similar, but non-identical," works—*e.g.*, Song A and Song A₁—can produce the "same output identifier." Pet. 23.



"Identifying a work" based on a hash of that "same output identifier" is then

performed using an exact match search (rather than a neighbor search) using “[a] table look-up using the identifier decoded from the electronic work.” Pet. 23. That the hashed output identifier is compared to the records using an exact match lookup is necessary because, as explained above, there is no correlation between the pre-hashed songs and the hashed identifiers because hashing eliminates any such relationship. This is because the output of hash functions is dramatically changed even if one bit of input changed. Ex. 2003, 2:1-10.

Levy (incorporated by reference) confirms that although there may be slight differences in two versions of an original work (*e.g.*, A and A₁), the two versions can be pre-processed to eliminate such difference (*e.g.*, using “a low pass filter”) to create the exact same identifier to be hashed and compared:

One component of fingerprint algorithm is a hash algorithm. The hash algorithm may be applied to a selected portion of a music file (*e.g.*, the first 10 seconds) to create a fingerprint. It may be applied to discrete samples in this portion, or to attributes that are less sensitive to typical audio processing. Examples of less sensitive A attributes include most significant bits of audio samples or a low pass filtered version of the portion.

Levy, 9:52-61. As a result, Levy (incorporated into Conwell) teaches that similar data can be pre-processed (*e.g.*, using a low pass filter) to generate signatures that are identical before hashing and, as a result, the hashed output would also be

identical.

Petitioner presents no reasoning or evidence demonstrating that Conwell discloses a “neighbor” search because a “neighbor” search is inconsistent with the fact that, for hashes, “even a single bit difference results in an entirely different hash value.” Ex. 2003, 2:1-10.

* * *

Ground 2 should not be instituted for the reasons set forth above.

VI. Ground 3: Anticipated by Wood (Ex. 1015) fails.

A. Overview of Wood.

Wood discloses “a method for recognizing a musical composition from a specimen that is provided by a customer (as by humming, singing, or otherwise vocalizing the specimen or by picking it out on a simulated piano or other tone generator).” Wood, 1:13-20. As illustrated in Figure 2, a user can “hum or otherwise vocalize” the desired song (“specimen”) into a “microphone 42.” Wood, Abstract; Wood, 1:13-20. “A music-recognition system then locates candidates for the selected composition and displays identifiers for these candidates to the customer.” Wood, Abstract. Wood discloses that “a distant score is calculated between the specimen and each entry in the pattern library.” Wood, 6:8-23.

B. Ground 3 fails because Wood does not disclose a “non-exhaustive neighbor search.”

Each independent claim of the ‘441 Patent includes the phrase “non-exhaustive neighbor search.”

Wood does not disclose that the search algorithm used to identify matching candidates based on the specimen is a “non-exhaustive” search. Wood discloses that “a distant score is calculated between the specimen and each entry in the pattern library.” Wood, 6:8-23. Because the algorithm used in the music-recognition algorithm disclosed in Wood is used to evaluate “each entry in the pattern library” (rather than using intelligence to search a subset of the records the library), the search is exhaustive rather than non-exhaustive. *See also* Wood, 10:56-11:19 (confirming that a specimen is “checked against the patterns in the library 88,” not some portion of the patterns in the library 88).

The Petition (and corresponding declaration) fails to apply its construction of “non-exhaustive neighbor search” (or any construction) to Wood. Accordingly, the Petition fails to satisfy its burden of demonstrating that Wood teaches a “non-exhaustive neighbor search” because it fails to “map the construed claim language to the teachings of the asserted prior art references.” *International Securities*, IPR2014-00097, Paper 12 at 26.

As support for the claimed “non-exhaustive” search, the Petition (and

corresponding declaration) relies on two assertions. Both are taken out of context and do not fall within the scope of even Petitioner's improper construction of "non-exhaustive." Each assertion is addressed in turn.

Petitioner's Assertion (1): "Wood further discloses that 'the speed of the pattern matching could be increased considerably' by performing a non-exhaustive search that compares 'only . . . portions of each composition, possibly the chorus and one or two other memorable stretches of music.'" Pet. 36 (citing Wood, 11:28-42. This passage from Wood does not disclose a "non-exhaustive" search for two reasons.

(1) As demonstrated above, a "non-exhaustive" search does not require comparing all data within each potential match; just comparing each potential match. *See* section III(A). Accordingly, had Wood actually disclosed that the composition to be identified is compared to only a portion of the data for each composition in the reference database, this would still be an exhaustive search (rather than the claimed "non-exhaustive" search) because "each composition" is considered.

(2) Even under Petitioner's improper characterization of "non-exhaustive," this cited passage from Wood does not teach a "non-exhaustive" search, as demonstrated by reading the full passage from Wood:

Although it would be possible for pattern library 88 to store pitch and duration sequences for the entirety of each composition from beginning to end, the speed of the pattern matching could be increased considerably if the pattern library 88 only includes portions of each composition, possibly the chorus and one or two other memorable stretches of music that might tend to stick in the mind of consumers. Wood, 11:28-34.

Wood does not teach that only a portion of the data in the pattern library (rather than all data) is compared. Rather, Wood states that what is “store[ed]” in the pattern library could only consist of “portions of each composition:” “it is sufficient for the pattern library to store only normalized features for popular portions of each song.” Wood, 6:5-8. The work to be identified is then compared to the all the data in the entire library comprising portions of stored.

Petitioner's Assertion (2): “Alternatively, Wood discloses a non-exhaustive search that ‘ignore[s] periods of silence in the specimen.’” Pet. 36 (citing Wood, 8:5-12; 6:3-7). As demonstrated by the context of Wood from which Petitioner pulled this passage, Wood states that, when creating the fingerprint of the specimen, the disclosed system might ignoring periods of silence in the specimen (*i.e.*, the person humming the song to be recognized), not ignoring data in the pattern library when performing a comparison. This could be useful because “many customers can be expected to vocalize their specimen using a string of

dummy words, such as 'da-da-da,' leaving pauses of periods of substantially reduced volume even though the composition they are audibilizing may lack such pauses or periods of reduced volume.” Wood, 7:59-64. Wood states that the periods of silence improve accuracy during extraction fingerprinting stage, not the comparison stage:

ignoring periods of silence in the specimen will be beneficial since this will accommodate differences in the way that customers vocalize music (particularly when using dummy words at different frequencies with low volume or no volume between the dummy words), while ignoring rests will not substantially diminish the utility of the patterns that are produced.

Wood, 8:5-12. The periods of silence are used in the extraction step (creating the fingerprint), not the comparison step.

* * *

Ground 3 should not be instituted for the reasons set forth above.

VII. Ground 4: Obvious over Ghias (Ex. 1010) in view of Philyaw (Ex. 1014) fails.

A. Overview of Ghias.

Ghias discloses “an apparatus [for] searching melodies.” Ghias, Abstract. As illustrated in Figure 1, a “tune 12 is hummed by a person 18 into a microphone 20.” Ghias, 2:41-42. The data from the microphone is fed into “a pitch tracking

module 22 in computer 16” which extracts “a contour representation” of the melody (23). Ghias, 2:46-49. The computer uses a “query engine 24” which “searches the melody database 14.” Ghias, 2:50-59.

The search process disclosed in Ghias is exhaustive rather than “non-exhaustive.” The “query engine 24” compares the work (user input 23) to “all the songs” in the melody database 14 (the library). Ghias, 5:66-6:2. After searching all possible matches, the system “output[s] a ranked list of approximately matching melodies.” Ghias, 2:50-53.

B. Overview of Philyaw.

Philyaw discloses a system that uses identifying information embedded into a broadcast signal to view corresponding information. Philyaw, Abstract; 1:66-2:8.

A program is broadcasted having embedded therein a routing signal having routing information contained therein. The routing signal is then extracted from the broadcast. Thereafter, a personal computer is controlled to allow a user to retrieve the information from a storage region at the defined location, which defined location is located with the extracted routing information, providing it at the personal computer for use by the user.

Philyaw, 2:1-9.

Philyaw does not disclose a searching algorithm. The Petition does not assert that Philyaw discloses the claimed “non-exhaustive neighbor search” but

instead exclusively relies on Ghias for this element in ground 4 (the only ground in which Philyaw is asserted). Pet. 47-48; 51.

C. Ground 4 fails because the Petition relies on Ghias for a “non-exhaustive neighbor search” and Ghias does not disclose this claimed element.³

All independent claims of the ‘441 Patent include the limitation “non-exhaustive neighbor search.” ‘441, claims 1, 13, and 25. For ground 4, Petitioner relies exclusively on Ghias for the claimed “non-exhaustive neighbor search.” Pet. 49-60. Ground 4 fails because Ghias does not disclose the claimed “non-exhaustive neighbor search.”

Ghias teaches an exhaustive search that compares the work (user input 23) with “all the songs” in the database (*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. The user input (23) is not compared with some songs in melody database (14) but rather it “is compared with all the songs.”

³ If Trial were instituted, Network-1 would demonstrate that Ghias and Philyaw teach away from the combination of ground 4.

Petitioner asserts that the explicit statement that the work “is compared with all the songs” does not apply to the search disclosed in Ghias but instead “refers to preprocessing that occurs at the time new works are added to a database to facilitate later search, not search itself.” Ex. 1004 ¶121. Petitioner’s argument fails for two reasons.

First, to make this argument, Petitioner ignores the first part of the sentence (omitted in Petitioner’s declaration when quoting this passage from Ghias) confirming that “all the songs” refers to the “search” (and not to preprocessing): “In order to search the database” Ghias, 5:66-6:2.

Second, Petitioner cannot explain why the pre-processed data would be “compared with all the songs” in the database to generate the database. There is no reason or teaching of a need to compare pre-processed data in the database to itself.

Ghias does not address or consider the concept of trying to optimize a search by limiting it to only a subset of records in the database. Rather, Ghias addresses efficiencies in the comparison of the work to each record in the database—an efficient way of comparing a melody (the work) with a given reference melody (a record in the database). Once that single comparison is performed, the system moves on to the next reference melody and continues comparing “all the songs” in the reference database. Ghias, 5:66-6:2. Accordingly, Ghias teaches a brute force

exhaustive algorithm (rather than the claimed “non-exhaustive” search) that applies no intelligence to the process of selecting the melody references in melody database to be compared to the melody to be identified.

As support that Ghias discloses the claimed “non-exhaustive” search, the Petition (and corresponding declaration) rely on the following assertion:

Ghias further discloses that this search may be non-exhaustive: ‘it is considered desirable to use an efficient approximate pattern matching algorithm’ rather than an algorithm that is guaranteed to yield a match. *Id.* at 6:7-11; 6:23-35 (‘Several Algorithms have been developed . . . Running times have ranged from $O(mn)$ for the brute force algorithm to $O(kn)$ or $O(n\log(m))$ ’).

Pet. 47-48; 51; Ex. 1004 ¶ 122.

This assertion fails to: (a) apply Petitioner’s construction (or any other construction) to Ghias; or (b) explain how an “efficient approximate pattern matching algorithm” is expressly or inherently a “non-exhaustive” search.

Accordingly, the Petition fails to satisfy its burden because it fails to “map the construed claim language to the teachings of the asserted prior art references.”

International Securities, IPR2014-00097, Paper 12 at 26.

Moreover, the Petition must quote Ghias out of context to try to establish that Ghias discloses concepts similar to Petitioner's proposed construction. The full context is depicted at

right. Ghias, 6:3-28.

These passages discuss comparing the work with a single record in the database. The

“approximate string matching” algorithms

discussed in these

passages involve

matching a work with a

record in the database that includes an “error” so that “various forms of errors”

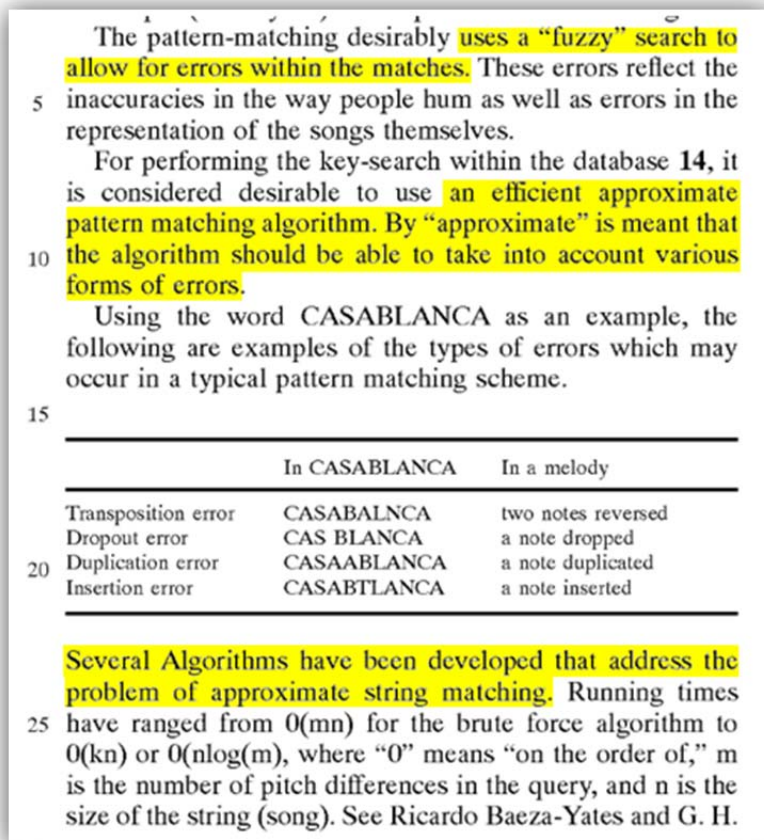
would not prevent a proper match from being identified. *Id.* In Ghias, the

“approximate string matching” algorithm is applied when the work melody “is

compared with all the songs” in the database (Ghias, 5:66-6:2). Accordingly,

Ghias does not disclose a search that would even meet Petitioner's improper

construction of “non-exhaustive search,” because Ghias does not search less than



“all possible matches” or even not “all data within all possible matches.” Pet. 6.

* * *

Ground 4 should not be instituted for the reason set forth above.

VIII. Conclusion.

The Petition should be denied because each of the four grounds presented in the Petition fails.

Respectfully submitted,

AMSTER, ROTHSTEIN & EBENSTEIN LLP
Attorneys for Patent Owner
90 Park Avenue
New York, NY 10016
(212) 336-8000

Dated: March 27, 2015

By: /Charles R. Macedo/
Charles R. Macedo
Registration No. 32,781

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), the undersigned hereby certifies that on this 27th day of March, 2015, a copy of the foregoing PATENT OWNER'S PRELIMINARY RESPONSE TO THE PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 8,656,441 UNDER 35 U.S.C. § 313 PURSUANT TO 37 C.F.R. § 42.107, together with Patent Owner's Exhibit List and Exhibit Nos. 2001-2003, was served via e-mail on the counsel of record for Petitioner at the following e-mail addresses:

DLGOOGLE1N@skadden.com

James J. Elacqua (James.Elacqua@skadden.com)

Douglas R. Nemec (Douglas.Nemec@skadden.com)

Dated: March 27, 2015

By: /Charles R. Macedo/
Charles R. Macedo
Registration No. 32,781