ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220723096

A Survey of Music Information Retrieval Systems

Conference Paper · January 2005

Source: DBLP

CITATIONS
242

READS
2,353

3 authors, including:

Rainer Typke
European Commission
27 PUBLICATIONS **861** CITATIONS

SEE PROFILE

Remco C. Veltkamp
Utrecht University
317 PUBLICATIONS **8,553** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

AIM@SHAPE - Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects View project

Project

Healthy play, better coping: The importance of play for the development of children in health and disease View project

All content following this page was uploaded by Remco C. Veltkamp on 27 May 2014.

The user has requested enhancement of the downloaded file.

A SURVEY OF MUSIC INFORMATION RETRIEVAL SYSTEMS

Rainer Typke, Frans Wiering, Remco C. Veltkamp
Universiteit Utrecht
Padualaan 14, De Uithof
3584CH Utrecht, The Netherlands
rainer.typke,frans.wiering,remco.veltkamp@cs.uu.nl

ABSTRACT

This survey paper provides an overview of content-based music information retrieval systems, both for audio and for symbolic music notation. Matching algorithms and indexing methods are briefly presented. The need for a TREC-like comparison of matching algorithms such as MIREX at ISMIR becomes clear from the high number of quite different methods which so far only have been used on different data collections. We placed the systems on a map showing the tasks and users for which they are suitable, and we find that existing content-based retrieval systems fail to cover a gap between the very general and the very specific retrieval tasks.

Keywords: MIR, matching, indexing.

1 INTRODUCTION

This paper gives an overview of Music Information Retrieval (MIR) systems for content-based music searching, preceded by a brief overview of the methods commonly used by these systems. Unlike the existing literature (Downie, 2003; Birmingham et al., 2003), we try to place the systems on a two-dimensional map of retrieval tasks and targeted users. Information about the systems was collected with the help of a website (http://mirsystems.info) with a questionnaire where developers of MIR systems can enter descriptions of their systems, including publications, matching methods, features, indexing method, and collection size. Most of the information in this paper, however, comes from the publications containing the developers' own evaluations of their systems.

Two main groups of MIR systems for content-based searching can be distinguished, systems for searching audio data and systems for searching notated music. There are also hybrid systems that first convert audio signal into a symbolic description of notes and then search a database of notated music.

Content-based music search engines can be useful for a variety of purposes and audiences:

- Query-by-Humming: in record stores, it is not uncommon for customers to only know a tune from a record they would like to buy, but not the title of the work, composer, or performers. Salespeople with a vast knowledge of music who are willing and able to identify tunes hummed by customers are scarce, and it could be interesting to have a computer do the task of identifying melodies and suggesting records.
- A search engine that finds musical scores similar to a given query can help musicologists find out how composers influenced one another or how their works are related to earlier works of their own or by other composers. This task has been done manually by musicologists over the past centuries. If computers could perform this task reasonably well, more interesting insights could be gained faster and with less effort.
- Copyright issues could be resolved, avoided or raised more easily if composers could easily find out if someone is plagiarizing them or if a new work exposes them to the risk of being accused of plagiarism.

Content-based search mechanisms that work specifically for audio recordings can be useful for the following purposes:

- It is possible to identify music played, for example, on the radio or in a bar by pointing a cellular phone at the speakers for a few seconds and using an audio fingerprinting system for identifying the exact recording that is being played.
- Recordings made by surveillance equipment can be searched for suspicious sounds.
- Content-based video retrieval can be made more powerful by analyzing audio content.
- Theaters, film makers, and radio or television stations might find a search engine useful that can find sound effects similar to a given query or according to a given description in a vast library of audio recordings.

Although MIR is a rather young field, and the problems of MIR are challenging (Byrd and Crawford, 2002),there

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

©2005 Queen Mary, University of London

are already commercial applications of MIR systems. The automatic identification of recordings via cellular phones using audio fingerprinting, for example, is offered by Shazam[1], a UK-based service that charges its customers for identifying tunes and also offers matching ringtones and CDs.

2 SEARCHING SYMBOLIC DATA

2.1 String-based methods for monophonic melodies

Monophonic music can be represented by one-dimensional strings of characters, where each character describes one note or one pair of consecutive notes. Strings can represent interval sequences, gross contour, sequences of pitches and the like, and well-known string matching algorithms such as algorithms for calculating editing distances, finding the longest common subsequence, or finding occurrences of one string in another have been applied, sometimes with certain adaptations to make them suitable for matching melodies.

2.1.1 Distance Measures

Some MIR systems only check for exact matches or cases where the search string is a substring of database entries. For such tasks, standard string searching algorithms like Knuth-Morris-Pratt and Boyer-Moore can be used. Themefinder (see Section 4.17) searches the database for entries matching regular expressions. In this case, there is still no notion of distance, but different strings can match the same regular expression.

For approximate matching, it can be useful to compute an editing distance with dynamic programming. Musipedia is an example of a system that does this (see Section 4.7). Simply computing an editing distance between query strings and the data in the database is not good enough, however, because these strings might represent pieces of music with different lenghts. Therefore, it can be necessary to choose suitable substrings before calculating an editing distance.

2.1.2 Indexing

For finding substrings that match exactly, the standard methods for indexing text can be used (for example, inverted files, B-trees, etc.). The lack of the equivalent of words in music can be overcome by just cutting melodies into n-grams (Downie, 1999) and indexing those.

For most editing distances that are actually useful, the triangle inequality holds[2]. Therefore, the vantage indexing method described in Typke et al. (2003) can be used for those, but other methods like metric trees or vantage point trees are also possible.

2.2 Set-based methods for polyphonic music

Unlike string-based methods, set-based methods do not assume that the notes are ordered. Music is viewed as a set of events with properties like onset time, pitch, and duration.

2.2.1 Distance Measures

Clausen et al. (2000) proposed a search method that views scores and queries as sets of notes. Notes are defined by note onset time, pitch, and duration. Exact matches are supersets of queries, and approximate matching is done by finding supersets of subsets of the query or by allowing alternative sets.

Typke et al. (2003) also view scores and queries as sets of notes, but instead of finding supersets, they use transportation distances such as the Earth Mover's Distance for comparing sets (see 4.9).

2.2.2 Indexing

By quantizing onset times and by segmenting the music into measures, Clausen et al. (2000) make it possible to use inverted files. Typke et al. (2003) exploit the triangle inequality for indexing, which avoids the need for quantizing. Distances to a fixed set of vantage objects are precalculated for each database entry. Queries then only need to be compared to entries with similar distances to the vantage objects.

2.3 Probabilistic Matching

The aim of probabilitstic matching methods is to determine probabilistic properties of candidate pieces and compare them with corresponding properties of queries. For example, the GUIDO system (see Section 4.5) calculates Markov models describing the probabilities of state transitions in pieces and then compares matrices which describe transition probabilities.

2.3.1 Distance Measures

Features of melodies such as interval sequences, pitch sequences, or rhythm can be used to calculate Markov chains. In these Markov chains, states can correspond with features like a certain pitch, interval, or note duration, and the transition probabilities reflect the numbers of occurrences of different subsequent states. The similarity between a query and a candidate piece in the database can be determined by calculating the product of the transition probabilities, based on the transition matrix of the candidate piece, for each pair of consecutive states in the query. See Section 4.5 for an example of a MIR system with probabilistic matching.

2.3.2 Indexing: Hierarchical Clustering

Transition matrices can be organized as a tree. The leaves are the transition matrices of the pieces in the database, while inner nodes are the transition matrices describing the concatenation of the pieces in the subtree. See Section 4.5 or Hoos et al. (2001) for a more detailed description.

[1] `http://www.shazam.com`, not to be confused with `http://www.shazam.co.uk`

[2] An example for a not very useful editing distance would be one where any character can be replaced with one special character at no cost. That way, the detour via a string consisting only of that special character would always yield the distance zero for unequal strings of the same length.

3 SEARCHING AUDIO DATA

3.1 Extracting perceptionally relevant features

A natural way of comparing audio recordings in a meaningful way is to extract an abstract description of the audio signal which reflects the perceptionally relevant aspects of the recording, followed by the application of a distance function to the extracted information. An audio recording is usually segmented into short, possibly overlapping frames which last short enough such that there are not multiple distinguishable events covered by one frame. Wold et al. (1996) list some features that are commonly extracted from audio frames with a duration between 25 and 40 milliseconds:

- **Loudness:** can be approximated by the square root of the energy of the signal computed from the short-time Fourier transform, in decibels.
- **Pitch:** the Fourier transformation of a frame delivers a spectrum, from which a fundamental frequency can be computed with an approximate greatest common divisor algorithm.
- **Tone (brightness and bandwidth):** Brightness is a measure of the higher-frequency content of the signal. Bandwidth can be computed as the magnitude-weighted average of the differences between the spectral components and the centroid of the short-time Fourier transform. It is zero for a single sine wave, while ideal white noise has an infinite bandwidth.
- **Mel-filtered Cepstral Coefficients** (often abbreviated as MFCCs) can be computed by applying a mel-spaced set of triangular filters to the short-time Fourier transform, followed by a discrete cosine transform. The word “cepstrum” is a play on the word “spectrum” and is meant to convey that it is a transformation of the spectrum into something that better describes the sound characteristics as they are perceived by a human listener. A mel is a unit of measure for the perceived pitch of a tone. The human ear is sensitive to linear changes in frequency below 1000 Hz and logarithmic changes above. Mel-filtering is a scaling of frequency that takes this fact into account.
- **Derivatives:** Since the dynamic behaviour of sound is important, it can be helpful to calculate the instantaneous derivative (time differences) for all of the features above.

Audio retrieval systems such as the system described in Section 4.16 compare vectors of such features in order to find audio recordings that sound similar to a given query.

3.2 Audio Fingerprinting

If the aim is not necessarily to identify a work, but a recording, audio fingerprinting techniques perform quite well. All phone-based systems for identifying popular music (e. g., Shazam) use some form of audio fingerprinting. A feature extractor is used to describe short segments of recordings in a way that is as robust as possible against the typical distortions caused by poor speakers, cheap microphones, and a cellular phone connection, as well as background noise like people chatting in a bar. Such features do not need to have anything to do with human perception or the music on the recording, they just need to be unique for different recordings and robust against distortions. These audio fingerprints, usually just a few bytes per recording segment, are then stored in a database index, along with pointers to the recordings where they occur. The same feature extractor is used on the query, and with the audio fingerprints that were extracted from the query, candidates for matching recordings can be quickly retrieved. The number of these candidates can be reduced by checking whether the fingerprints occur in the right order and with the same local timing.

3.3 Set-based Methods

Clausen and Kurth used their set-based method (see Section 2.2) also for audio data. They use a feature extractor for converting PCM[3] signals into sets that can be treated the same way as sets of notes.

3.4 Self-Organizing Map

Self-Organizing Map (SOM), a very popular artificial neural network algorithm in the unsupervised learning category, has been used for clustering similar pieces of music and classifying pieces, for example by Rauber et al. (2003). Section 4.14 describes their system, which extracts feature vectors that describe rhythm patterns from audio, and clusters them with a SOM.

4 MIR SYSTEMS

Table 1 gives an overview of the characteristics of 17 MIR systems. The following subsections contain additional information about these systems.

4.1 audentify!

URL: `http://www-mmdb.iai.uni-bonn.de/eng-public.html`
The fingerprints are sequences of bits with a fixed length, where every bit describes one audio window. The collection contains about 15.000 MP3 files (@128kBit/s), approx. 1.5 month of audio data.
Literature: Kurth et al. (2002b), Kurth et al. (2002a), Ribbrock and Kurth (2002), Kurth (2002), Clausen and Kurth (2002)

4.2 C-Brahms

URL: `http://www.cs.helsinki.fi/group/cbrahms/demoengine/`
C-Brahms employs nine different algorithms called P1, P2, P3, MonoPoly, IntervalMatching, ShiftOrAnd, PolyCheck, Splitting, and LCTS offering various combinations of monophony, polyphony, rhythm invariance, transposition invariance, partial or exact matching.

Literature: Ukkonen et al. (2003), Lemström and Tarhio (2003), Lemström et al. (2003)

[3] PCM (Pulse Code Manipulation): raw uncompressed digital audio encoding.

Table 1: Content-based Music Information Retrieval systems.

	Input		Matching					Features									
Name	Audio	Symbolic	Audio	Symbolic	Exact	Approximate	Polyphonic	Audio Fingerprints	Pitch	Note Duration	Timbre	Rhythm	Contour	Intervals	Other	Indexing	Collection Size (Records)
audentify!	●		●			●	●	●								Inverted files	15,000
C-Brahms		●		●	●	●	●		●	●		●		●		none	278
CubyHum	●			●		●								●		LET	510
Cuidado	●		●			●	●				●	●			●	not described	works for $> 100,000$
GUIDO/ MIR		●		●		●			●	●		●		●	●	Tree of transition matrices	150
Meldex/ Greenstone	●	●		●		●							●	●		none	9,354
Musipedia	●	●		●		●							●			Vantage objects	$> 30,000$
notify! Whistle	●	●		●		●			●			●				Inverted files	2,000
Orpheus		●		●		●	●		●	●		●		●		Vantage objects	476,000
Probabilistic "Name That Song"		●		●		●								●	●	Clustering	100
PROMS		●		●	●	●			●			●				Inverted files	12,000
Cornell's "QBH"	●			●		●							●			none	183
Shazam	●		●		●		●	●								Fingerprints are indexed	$>$ 2.5 million
SOMeJB	●		●			●	●								●	Tree	359
SoundCompass	●			●		●			●			●				Yes	11,132
Super MBox	●			●		●			●			●				Hierarchical Filtering	12,000
Themefinder		●		●	●				●				●	●		none	35,000

4.3 CubyHum

Edit distances of one-dimensional pattern sequences (here: pitch intervals) are calculated. Nine interval classes are used; intervals above 6 semitones are not distinguished. Filtering is done with the LET algorithm (Chang and Lawler, 1994) with some heuristic adjustments. CubyHum still looks at every single database entry in every search. **Literature:** Pauws (2002)

4.4 Cuidado Music Browser

Besides similarity measures based on intrinsic features such as rhythm, energy, and timbre, there are also similarity measures based on metadata. A co-occurrence matrix keeps track of similar contexts like a radio program, album playlist, or web page. The authors do not describe an indexing method.

Literature: Pachet (2003), Pachet et al. (2003b), Pachet et al. (2003a)

4.5 GUIDO/MIR

URL: `http://www.informatik.tu-darmstadt.de/AFS/GUIDO/index.html`

Queries are a combination of melodic (absolute pitch, intervals, interval types, interval classes, melodic trend) and rhythmic information (absolute durations, relative durations, trend). First-order Markov chains are used for modeling the melodic and rhythmic contours of monophonic pieces of music. There is one Markov chain for each piece and each melodic or rhythmic query type. The states of these chains correspond with melodic or rhythmic features.

Transition matrices are organized as a tree (leaves: pieces; inner nodes: transition matrices describing the concatenation of the pieces in the subtree) with the aim of ruling out data with transition probabilities of zero at an early stage of the search, and heuristically guiding the search.
Literature: Hoos et al. (2001)

4.6 Meldex/Greenstone

URL: `http://www.nzdl.org/fast-cgi-bin/music/musiclibrary`
Meldex uses two matching methods: Editing distance calculation with dynamic programming and a state matching algorithm for approximate searching (Wu and Manber, 1992). The folk song collection is based on the Essen and Digital Tradition collections.
Literature: McNab et al. (May 1997), Bainbridge et al. (2004)

4.7 Musipedia

URL: `http://musipedia.org`
The search engine retrieves the closest 100 entries according to the editing distance of gross contour strings. The collection can be edited and expanded by any user. For indexing, the vantage object method described by Typke et al. (2004) is used for the first 6 characters of the contour string. Musipedia was known as "Tuneserver" in an earlier development state.
Literature: Prechelt and Typke (2001)

4.8 notify! Whistle

URL: `http://www-mmdb.iai.uni-bonn.de/projects/nwo/index.html`
Monophonic queries are matched against polyphonic sets of notes. A rhythm tracker enables matching even if there are fluctuations or differences in tempo. The audio queries can be symbolically edited in pianoroll notation.
Literature: Kurth et al. (2002a)

4.9 Orpheus

URL: `http://give-lab.cs.uu.nl/orpheus/`

Queries can be polyphonic. Notes are represented as weighted points in the 2-dimensional space of onset time and pitch. The Earth Mover's Distance or variants of it are used for calculating distances. For indexing, vantage objects are used.
Literature: Typke et al. (2003), Typke et al. (2004)

4.10 Probabilistic "Name That Song"

This system uses not only music, but also lyrics for matching. All note transitions and words from the query must occur at least once in a piece for it to be considered a match. The pieces in the database are clustered. The probability of sampling is computed for each cluster. A query is then performed in i iterations. In each iteration, a cluster is selected and the matching criteria are applied to each piece in this cluster until a match is found, which then becomes the rank-ith result.

The clustering prevents the algorithm from visiting every single piece in the database.
Literature: Brochu and de Freitas (2002)

4.11 PROMS

URL: `http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/`
PROMS views database entries and queries as sets of notes. Matches are supersets of queries. Queries can be fuzzy (a set of finite, nonempty sets of possible notes instead of a set of notes).

PROMS relies on measure information for segmenting and quantizes pitches and onset times. This makes it possible to use inverted files.
Literature: Clausen et al. (2000)

4.12 Cornell's "Query by Humming"

URL: `http://www.cs.cornell.edu/Info/Faculty/bsmith/query-by-humming.html`
After pitch tracking with autocorrelation, maximum likelihood, or cepstrum analysis, the gross contour is encoded with the alphabet U/D/S (up/down/same). The Baeza-Yates/Perleberg pattern matching algorithm is then used for finding all instances of a pattern string in a text string so that there are at most k mismatches.
Literature: Ghias et al. (1995)

4.13 Shazam

URLs: `http://www.shazam.com`, `http://ismir2003.ismir.net/presentations/Wang.PDF`
Audio fingerprints describe the relative time and pitch distances of future peaks within a fixed-size target zone for a given peak in the spectrum ("landmark"). For all database entries with fingerprints that match some fingerprints in the query, it is checked whether they occur at the correct relative times and at the correct landmarks. This method is very robust against noise and distortion caused by using a mobile phone connection and added background noise.
Literature: Wang (2003)

4.14 SOMeJB - The SOM-enhanced JukeBox

URL: `http://www.ifs.tuwien.ac.at/˜andi/somejb/`
A Self-Organizing Map (SOM) is used for clustering pieces. The SOM consists of units which are ordered on a rectangular 2-dimensional grid. A model vector in the high-dimensional data space is assigned to each of the units. During the training, the model vectors are fitted to the data such that the distances between the data items and the corresponding closest model vectors are minimized. Feature vectors contain amplitude values for selected frequency bands.

Training the neural network, i.e. the Growing Hierarchical Self-Organizing Map (GHSOM), an extension to the SOM, results in a hierarchical organization.

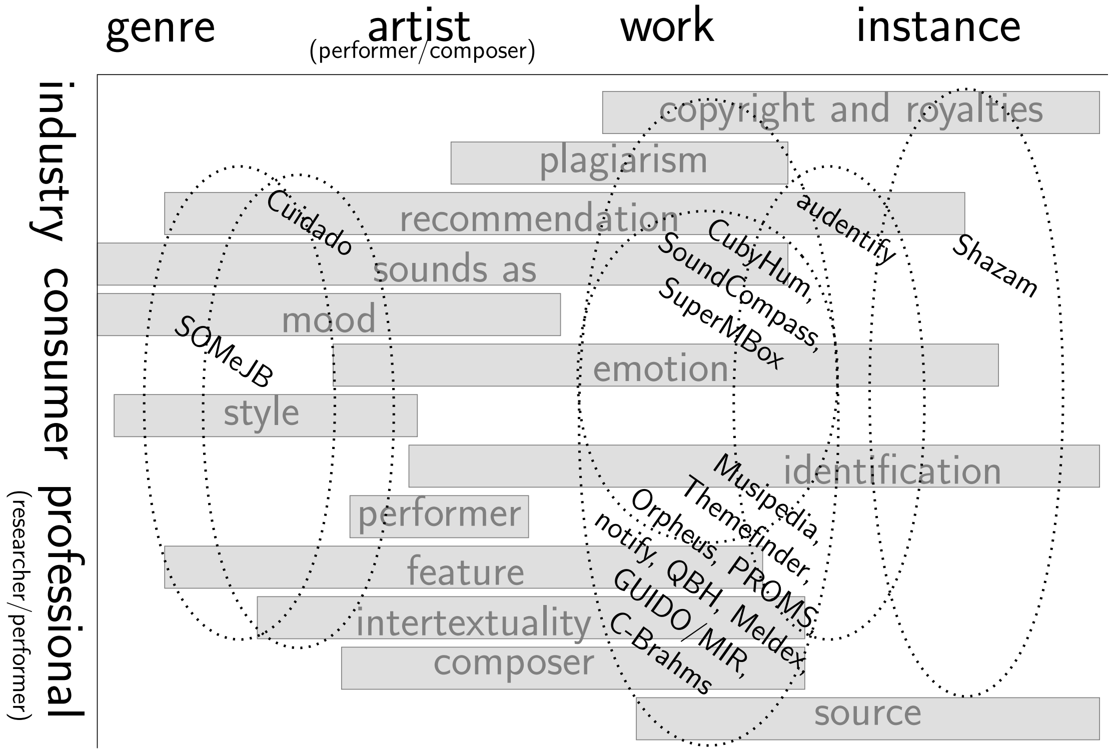

Figure 1: A mapping of MIR systems to retrieval tasks. See Section 5 for a discussion.

Literature: Rauber et al. (2003), Pampalk et al. (2002), Rauber et al. (2002b), Rauber et al. (2002a), Rauber and Frühwirth (2001)

4.15 SoundCompass

Users first set a metronome to a convenient tempo and then hum their melody so that the beats coincide with metronome clicks. Three feature vectors (Tone Transition, Partial Tone Transition, Tone Distribution) are stored for overlapping windows covering the songs (16 beats long, 4 beats apart from each other). Euclidean distance calculation, accelerated with an index.
Literature: Kosugi et al. (2000)

4.16 Super MBox

URL: `http://neural.cs.nthu.edu.tw/jang/demo/`
The acoustic input is converted into a pitch sequence with a time scale of 1/16 second. Dynamic time warping is used to compute the warping distance between the input pitch vector and that of every song in the database.
Literature: Jang et al. (2001)

4.17 Themefinder

URL: `http://themefinder.org`
Themefinder provides a web-based interface to the Humdrum thema command, which allows searching of databases containing musical themes or incipits with string matching algorithms.
Literature: Kornstädt (1998)

5 RETRIEVAL TASKS

In the introduction, we mentioned a number of MIR retrieval tasks. It is worthwhile to map the systems to these tasks. Three main audiences can be distinguished that can benefit from MIR:

1. industry: e. g. recording, broadcasting, performance
2. consumers
3. professionals: performers, teachers, musicologists

The level at which retrieval is needed may differ considerably:

1. work instance: the individual score or sound object
2. work: set of instances that are considered to be essentially the same
3. artist: creator or performer of work
4. genre: music that is similar at a very generic level, e. g. classical, jazz, pop, world music

This is not really a strict hierarchy. Artists perform in different genres, and one work can be performed, even created, by multiple artists. Also, there is rather a continuum. Genres can be divided into subgenres, artists grouped in schools. Even the “work” concept is not a fixed given. Beethoven’s Third Symphony, for example is determined by the composer’s score, and changing even one note can be a violation of the work, for example the famous “false entry” of the French Horn at the beginning of the recapitulation. On the other hand, different renditions of “I did it my way” are usually considered the same work even though the musical content may be rather different.

MIR retrieval tasks can be characterised by audience and level of retrieval. Often, tasks connect a subrange of the continuum (see Figure 1). A non-comprehensive overview of tasks (for typical search tasks and their frequencies of occurence, see also Lee and Downie (2004)) includes:

- copyright and royalties: receive payments for broadcast or publication of music
- detection of plagiarism: the use of musical ideas or stylistic traits of another artist under one's own name
- recommendation: find music that suits a personal profile
- sounds as: find music that sounds like a given recording
- mood: find music that suits a certain atmosphere
- emotion: find music that reflects or contradicts an emotional state
- style: find music that belongs to a generic category, however defined
- performer: find music by (type of) performer
- feature: employ technical features to retrieve works in a genre or by an artist
- composer: find works by one composer
- intertextuality: finding works that employ the same material or refer to each other by allusion
- identification: ascribing a work or work instance to an artist or finding works containing a given theme, query by humming
- source: identifying the work to which an instance belongs, for example because metadata are missing

Figure 1 shows how the MIR systems from Table 1 can be mapped to the tasks. Audio fingerprinting systems such as Shazam are particularly good at identifying recordings, that is, instances of works. This task must be based on audio information because in two different performances, the same music might be performed, and therefore only the audio information is different.

Audio data is also a good basis for very general identification tasks such as genre and artist. SOMeJB and Cuidado both use audio features for this purpose. Since it uses metadata, Cuidado can also cover tasks for which it helps to know the artist.

Query-by-humming systems such as SoundCompass, which is intended to be used in a Karaoke bar, make identification tasks easier for consumers who might lack the expertise that is needed for entering a sequence of intervals or a contour in textual form. These systems focus on identifying works or finding works that are similar to a query.

By offering the possibility of entering more complex queries, systems such as Themefinder, C-Brahms, and Musipedia cover a wider range of tasks, but they still can only be used on the work level. Since they work with sets of notes or representations that are based on sets of notes, they cannot be used for more specific tasks such as identifying instances, and their algorithms are not meant to do tasks on the more general artist and genre levels.

6 CONCLUSIONS

We probably covered only a small part of all existing MIR systems (we left some commercial systems out, for example MuscleFish's SoundFisher, because we could not find research papers about them), but we can still draw some conclusions from this survey.

A great variety of different methods for content-based searching in music scores and audio data has been proposed and implemented in research prototypes and commercial systems. Besides the limited and well-defined task of identifying recordings, for which audio fingerprinting techniques work well, it is hard to tell which methods should be further pursued. This underlines the importance of a TREC-like series of comparisons for algorithms (such as EvalFest/MIREX at ISMIR) for searching audio recordings and symbolic music notation.

Audio and symbolic methods are useful for different tasks. For instance, identification of instances of recordings must be based on audio data, while works are best identified based on a symbolic representation. For determining the genre of a given piece of music, approaches based on audio look promising, but symbolic methods might work as well.

Figure 1 shows that most MIR systems focus on the work level. There is a gap between MIR systems working on the genre level and those on the work level. Large parts of the more interesting tasks, such as specific recommendation, generic technical features, and intertextuality, fall into this gap. Using metadata might help cover this gap, but this would rule out the possibility of handling data for which the quality of known metadata is not sufficient. Manual annotation quickly gets prohibitively expensive. To fill the gap with completely automatic systems, it might be necessary to find algorithms for representing music at a higher, more conceptual abstraction level than the level of notes.

ACKNOWLEDGEMENTS

We thank everybody who filled in the questionnaire on `http://mirsystems.info`.

REFERENCES

D. Bainbridge, S. J. Cunningham, and J. S. Downie. Greenstone as a music digital library toolkit. In *ISMIR Proceedings*, pages 42–43, 2004.

W. Birmingham, C. Meek, K. O'Malley, B. Pardo, and J. Shifrin. Music information retrieval systems. *Dr. Dobb's Journal*, Sept. 2003.

E. Brochu and N. de Freitas. "Name That Song!": A probabilistic approach to querying on music and text. *NIPS.Neural Information Processing Systems: Natural and Synthetic*, 2002.

D. Byrd and T. Crawford. Problems of music information retrieval in the real world. *Information Processing and Management*, 38:249–272, 2002.

W. I. Chang and E. L. Lawler. Sublinear approximate string matching and biological applications. *Algorithmica*, 12(4/5):327–344, 1994.

M. Clausen, R. Engelbrecht, D. Meyer, and J. Schmitz. PROMS: a web-based tool for searching in polyphonic music. In *ISMIR Proceedings*, 2000.

M. Clausen and F. Kurth. A unified approach to content based and fault tolerant music identification. In *International Conference On Web Delivering of Music.*, 2002.

J. S. Downie. *Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text.* PhD thesis, University of Western Ontario, London, Ontario, Canada, 1999.

J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37:295–340, 2003.

A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming - musical information retrieval in an audio database. In *Proceedings ACM Multimedia*, 1995.

H. Hoos, K. Renz, and M. Görg. GUIDO/MIR - an experimental musical information retrieval system based on guido music notation. In *ISMIR Proceedings*, pages 41–50, 2001.

J.-S. Jang, H.-R. Lee, and J.-C. Chen. Super MBox: An efficient/effective content-based music retrieval system. In *9th ACM Multimedia Conference*, pages 636–637, 2001.

A. Kornstädt. Themefinder: A web-based melodic search tool. In W. Hewlett and E. Selfridge-Field, editors, *Melodic Similarity: Concepts, Procedures, and Applications, Computing in Musicology*, volume 11. MIT Press, Cambridge, 1998.

N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima. A practical query-by-humming system for a large music database. In *Proceedings ACM Multimedia*, pages 333–342, 2000.

F. Kurth. A ranking technique for fast audio identification. In *International Workshop on Multimedia Signal Processing.*, 2002.

F. Kurth, A. Ribbrock, and M. Clausen. Efficient fault tolerant search techniques for full-text audio retrieval. In *112th Convention of the Audio Engineering Society*, 2002a.

F. Kurth, A. Ribbrock, and M. Clausen. Identification of highly distorted audio material for querying large scale data bases. In *112th Convention of the Audio Engineering Society*, 2002b.

J. H. Lee and J. S. Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *ISMIR Proceedings*, pages 441–446, 2004.

K. Lemström, V. Mäkinen, A. Pienimäki, M. Turkia, and E. Ukkonen. The C-BRAHMS project. In *ISMIR Proceedings*, pages 237–238, 2003.

K. Lemström and J. Tarhio. Transposition invariant pattern matching for multi-track strings. *Nordic Journal of Computing*, 2003.

McNab, Smith, Bainbridge, and Witten. The New Zealand digital library MELody inDEX. *D-Lib Magazine*, May 1997.

F. Pachet. Content management for electronic music distribution. *CACM*, 46(4):71–75, 2003.

F. Pachet, A. Laburthe, and J.-J. Aucouturier. The Cuidado Music Browser: An end-to-end EMD system. In *Proceedings of the 3rd International Workshop on Content-Based Multimedia Indexing*, 2003a.

F. Pachet, A. Laburthe, and J.-J. Aucouturier. Popular music access: The Sony Music Browser. *Journal of American Society for Information Science*, 2003b.

E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proceedings of ACM Multimedia*, pages 570–579, 2002.

S. Pauws. CubyHum: a fully operational query by humming system. In *ISMIR Proceedings*, pages 187–196, 2002.

L. Prechelt and R. Typke. An interface for melody input. *ACM Transactions on Computer-Human Interaction*, 8 (2):133–149, 2001.

A. Rauber and M. Frühwirth. Automatically analyzing and organizing music archives. In *Proceedings of the 5. European Conference on Research and Advanced Technology for Digital Libraries*, Lecture Notes in Computer Science. Springer, 2001.

A. Rauber, E. Pampalk, and D. Merkl. Content-based music indexing and organization. In *Proceedings of the 25. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 409–410, 2002a.

A. Rauber, E. Pampalk, and D. Merkl. Using psychoacoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *ISMIR Proceedings*, pages 71–80, 2002b.

A. Rauber, E. Pampalk, and D. Merkl. The SOM-enhanced jukebox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research (JNMR)*, 32(2):193–210, 2003.

A. Ribbrock and F. Kurth. A full-text retrieval approach to content-based audio identification. In *International Workshop on Multimedia Signal Processing*, 2002.

R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In *ISMIR Proceedings*, pages 107–114, 2003.

R. Typke, R. C. Veltkamp, and F. Wiering. Searching notated polyphonic music using transportation distances. In *Proceedings of the ACM Multimedia Conference*, pages 128–135, New York, 2004.

E. Ukkonen, K. Lemström, and V. Mäkinen. Sweepline the music! *Computer Science in Perspective*, pages 330–342, 2003.

A. Wang. An industrial strength audio search algorithm. In *ISMIR Proceedings*, Baltimore, 2003.

E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

S. Wu and U. Manber. Fast text searching allowing errors. *CACM*, 35(10):83–89, 1992.