Content Based Music Retrieval

Remco C. Veltkamp, Frans Wiering, Rainer Typke
Department of Information and Computing Sciences, Utrecht University, Netherlands

Two main groups of Music Information Retrieval (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 transcribe audio signal into a symbolic description of notes and then search a database of notated music. An example of such music transcription is the work of Klapuri [10], which in particular is concerned with multiple fundamental frequency estimation, and musical metre estimation, which has to do with ordering the rhythmic aspects of music. Part of the work is based on known properties of the human auditory system.
Content-based music search systems can be useful for a variety of purposes and audiences:

- In record stores, customers may 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 (notations of musical works) 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 infringement could be resolved, avoided or raised more easily if composers could find out if someone is plagiarizing them or if a new work exposes them to the risk of being accused of plagiarism. A content-based music retrieval could facilitate such searches.

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, including music.
- Theatres, 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, there are already commercial applications of MIR systems. The automatic identification of recordings via cellular phones using audio fingerprinting, for example, is already offered by several companies that charge customers for identifying tunes and also offers matching ring tones and CDs.

Music Formats

We consider three basic representations of music: music notation, time-stamped events, and audio. Most music retrieval research is concerned with mainstream Western music, based on noted that have a definite pitch. See Byrd and Crawford [1] for a more elaborate overview of the challenges of music information retrieval for these formats.
Western music is notated in so-called Common Music Notation (CMN), which traces its origins to the Middle Ages. Basically, music scores in CMN represent time information horizontally and pitch vertically. Each musical event is rendered as a note that is placed on a staff consisting of 5 horizontal lines; the shape of the note indicates its relative duration (therefore they have names such as whole, half and quarter notes). Many additional symbols may be used: the primary purpose of CMN is to facilitate the

performance of a musical composition, by giving as detailed instructions as necessary for creating an adequate performance.
Musical Instrument Digital Interface, or MIDI, is a time stamped industry-standard protocol that defines musical events, allowing electronic musical instruments and computers to talk to each other [14]. Each pitch is stored as a time-stamped note-on and a note-off event. The format also allows saving information about tempo, key signatures, the names of tracks and instruments, and other information.
Digital audio comes in many different file and representation formats. Typically, audio files contain information about the resolution, sampling rate and type of compression. Groups like MPEG have created open standards for compression, for example MP3, the MPEG Audio Layer 3. PCM (Pulse Code Modulation) is a common method of storing and transmitting uncompressed digital audio. Since it is a generic format, it can be read by most audio applications. PCM is the format used on audio CDs, and also a very common format for WAV files, the default file format for digital audio on Windows.
See figure 1 for a comparison of music formats. The columns 'Convert to lower format' and 'Convert to higher format' indicate how difficult it is to convert one format into the next lower or higher level automatically.

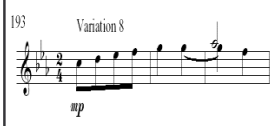
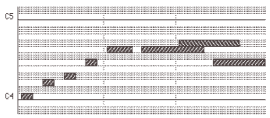
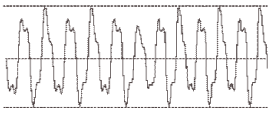
Music format	Example	Compared to image retrieval	Compared to text retrieval	Structure	Convert to lower format	Convert to higher format
music notation (Finale, Sibelius, MusicXML)		compound objects	text + markup	much	easy (OK job)	--
time-stamped events (MIDI)		objects, scenes	text	little	easy	fairly hard (OK job)
digital audio (MP3, Wav)		primitive features	speech	none	--	hard

Figure 1: Music format comparison, after [1].

Retrieval tasks

Several user groups exist for content-based music retrieval. Three main audiences can be distinguished that can benefit from MIR: (i) industry: e.g. recording, broadcasting, performance (ii) consumers (iii) professionals: performers, teachers, musicologists. The level at which retrieval is needed may differ considerably:

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

This is not a strict hierarchy. Artists perform in different genres, and one work can be performed, even created, by multiple artists. Also, this classification is a continuum rather than a nominal categorization. Genres can be divided into subgenres, artists grouped in schools. Even the "work" concept is not a fixed given. If a work is primarily determined by the composer's definitive score, changing even a single note

may be a violation of the work. On the other hand, different performances 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 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.

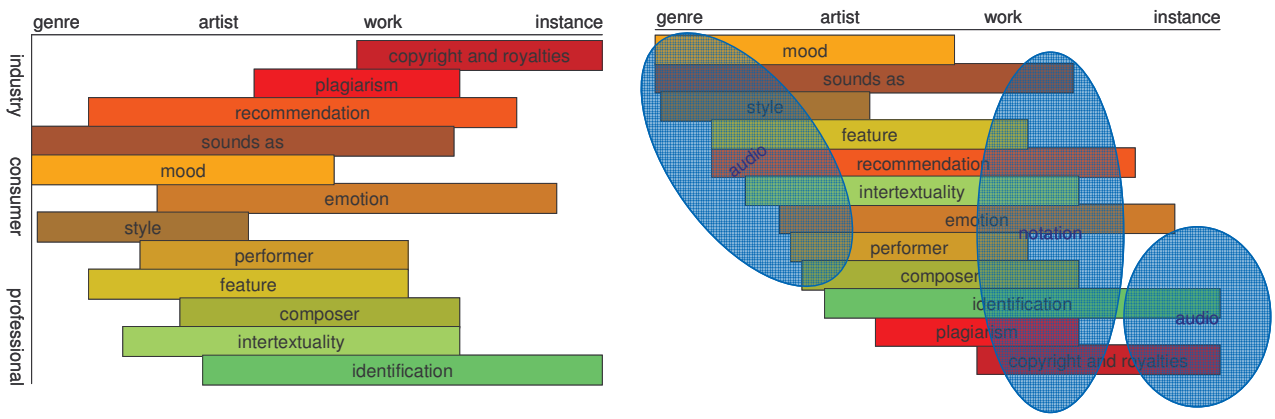


Figure 2: Music retrieval tasks. At the left the tasks are ordered by the type of users, at the right the tasks are ordered by the level of music retrieval.

For typical search tasks and their frequencies of occurrence, see also Lee and Downie [13]. Figure 2 shows which tasks are typical for the different user classes and level of music retrieval. Audio fingerprinting systems are particularly good at identifying recordings, that is, instances of works. This task must be based on audio information because in two different audio renditions, the same musical content 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 [19].

Query-by-humming systems 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.

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. By offering the possibility of entering more

complex queries, systems such as Themefinder [12] 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. How to address these levels is a research problem that is only beginning to be investigated. Music cognition and perception may have an important role in this.

Searching symbolic music

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.

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 [12] 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 [16]. 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 lengths. Therefore, it can be necessary to choose suitable substrings before calculating an editing distance.

More recently Cilibrasi et al. [3] have suggested using an approximation to Kolmogorov distance between two musical pieces as a means to compute clusters of music. They first process MIDI representation of a music piece to turn it into a string from a finite alphabet. Then they compute the distance between two music pieces using their Normalized Compression Distance (NCD). NCD uses the compressed length of a string as an approximation to its Kolmogorov complexity. The Kolmogorov complexity of a string is not computable, but the compressed length approximation gives good results for musical genre and composer clustering.

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 [6], where each N-gram is a sequence of N pitch intervals. For most editing distances that are actually useful, the triangle inequality holds. Therefore, indexing methods that rely on the triangle inequality property of the distance measure can be used, for example metric trees, vantage point trees, or the vantage indexing method described in [18].

Geometry-based methods for polyphonic melodies

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. Clausen et al. [4] propose 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. By quantizing onset times and by segmenting the music into measures, they make it possible to use inverted files.

Typke et al. [18] 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. They exploit the triangle inequality for indexing, which avoids the need for quantizing. Distances to a fixed set of vantage objects are pre-calculated for each database entry. Queries then only need to be compared to entries with similar distances to the vantage objects.

Ukkonen et al. [20] propose a number of algorithms for searching notated music. One method find translations of the query pattern such that all onset times and pitches of notes in the query match with some onset times and pitches of notes in the database documents. Another method finds translations of

the query pattern such that some onset times and pitches of the query match with some onset times and pitches of database documents. A third one finds translations of the query pattern that give longest common shared time (i.e., maximize the times at which query notes sound at the same time and with the same pitch as notes from the database documents). This algorithm does not take into consideration whether onset times match.

Probabilistic Matching

The aim of probabilistic matching methods is to determine probabilistic properties of candidate pieces and compare them with corresponding properties of queries. For example, the GUIDO system [9] calculates Markov models describing the probabilities of state transitions in pieces and then compares matrices which describe transition probabilities. 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. 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.

Searching musical audio

Most audio retrieval overviews (e.g. [8]) are focused on speech-oriented systems. In this chapter we will focus on musical audio. Clausen and Kurth [5] have used their geometry-based method (see above) also for musical audio data. They use a feature extractor for converting PCM3 signals into sets that can be treated the same way as sets of notes. Self-Organizing Map (SOM), have been used for clustering similar pieces of music and classifying pieces. For example [17] describes a system that extracts feature vectors that describe rhythm patterns from audio, and clusters them with a SOM.

Feature extraction

A natural way of comparing audio recordings 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 no multiple distinguishable events covered by one frame. Wold et al. [21] 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 shorttime 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 shorttime 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.[1] It transforms the spectrum into perception-based sound characteristics. 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.

[1] "Cepstrum" is a reordering of letters from "spectrum".

Other features include frequencies, attack (the duration from a zero to a maximum amplitude), decay (the duration from the initial maximum amplitude to a stable state amplitude), sustain (the level of the steady state amplitude), release (the duration from the steady state to its final zero amplitude), zero crossing rate, and spectral centroid (the average frequency, weighted by amplitude, of a spectrum). Many audio retrieval systems compare vectors of such features in order to find audio recordings that sound similar to a given query.
It is also possible to directly use coefficients used in the compression scheme. For example, [11] use the coefficients extracted from MP3 coded audio, representing the output of the polyphase filters used in MP3 compression. The polyphase filter bank divides the audio signal into 32 equal-width frequency subbands. Based on the human auditory system, the psychoacoustic model is designed for determining whether the coefficient of a subband should be coded.

Audio Fingerprinting

If the aim is not necessarily to identify a work, but a recording, audio fingerprinting techniques perform quite well. An audio fingerprint is a content-based compact signature that summarizes an audio recording [2]. All phone-based systems for identifying popular music 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. Common fingerprint requirements include [2]:

- Discriminative power over huge collections of fingerprints, so as to keep the number of false positives limited.
- Robustness to distortions such as additive noise and microphone distortions.
- Compactness for ease of processing.
- Computational simplicity for speed of processing.

Concluding Remarks

As can be seen from figure 2, there is a gap between the retrieval tasks and levels that musical audio and notated music are covering. It is a research challenge to fill this gap; a seminal project in this direction is the OMRAS project, http://www.omras.org. Further developments in this direction can be expected by integrating notation and audio based approaches into a high level symbolic approach, for example by audio transcription or harmonic matching [15].
For both audio and notated music, it is believed that retrieval performance may be greatly improved by using human-centered features, rather than technology-based features. Indeed, music is not so much perceived or remembered as a sequence of individual notes or frequencies. Music cognition and perception theory may play an important role in future retrieval systems.
Finally, a primary forum for music retrieval is the International Conference on Music Information Retrieval (ISMIR) series, http://www.ismir.net, and the Music Information Retrieval mailing list, http://listes.ircam.fr, but other music and multimedia related conference are also concerned with related issues.

See: [Here are the titles of derived short articles (font 11)]

References

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

2. P. Cano, E. Batlle, T. Kalker, J. Haitsma. A Review of Algorithms for Audio Fingerprinting. Proceedings of the International Workshop on Multimedia Signal Processing 2002.
3. R. Cilibrasi, P. Vitanyi, R. de Wolf. Algorithmic clustering of music based on string compression, Computer Music J., 28:4, 2004, 49-67.
4. M. Clausen, R. Engelbrecht, D. Meyer, and J. Schmitz. PROMS: a web-based tool for searching in polyphonic music. In ISMIR Proceedings, 2000.
5. 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.
6. 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.
7. A. Ghias et al., Query by Humming: Musical Information Retrieval in an Audio Database," In Proc. of Third ACM International Conference on Multimedia, 1995, p. 231-236.
8. J. Foote. An overview of audio information retrieval. Multimdia Systems, 7(1), 1999, p. 2-10.
9. H. Hoos, K. Renz, and M. Görg. GUIDO/MIR - an experimental musical information retrieval system based on GUIDO music notation. In ISMIR Proceedings, p. 41–50, 2001.
10. A. Klapuri. Signal Processing Methods for the Automatic Transcription of Music. PhD thesis, Tampere University of Technology, 2004.
11. C.-C. Liu, P.-J. Tsai. Content-based retrieval of MP3 music objects. In Proceedings of fthe 10th International Conference on Information and Knowledge Management, 2001, p. 506-511.
12. 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.
13. J. H. Lee and J. S. Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In ISMIR Proceedings, p. 441–446, 2004.
14. MIDI, Musical Instrument Digital Interface, www.midi.org
15. J. Pickens, J. P. Bello, G. Monti, T. Crawford, M. Dovey, M. Sandler, D. Byrd. Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach. In: Proceedings ISMIR 2002, 3rd International Conference on Music Information Retrieval.
16. L. Prechelt and R. Typke. An interface for melody input. ACM Transactions on Computer-Human Interaction, 8 (2):133–149, 2001.
17. A. Rauber, E. Pampalk, and D. Merkl. The SOMenhanced jukebox: Organization and visualization of music collections based on perceptual models. Journal of New Music Research (JNMR), 32(2):193–210, 2003.
18. R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In ISMIR Proceedings, p. 107–114, 2003.
19. G. Tzanetakis, Perry Cook. Musical Genre Classification of Audio Signals, *IEEE* Transactions on Speech and Audio Processing, 10(5), July 2002.
20. E. Ukkonen, K. Lemström, and V. Mäkinen (2003). Geometric Algorithms for Transposition Invariant Content-Based Music Retrieval. ISMIR 2003: Proceedings of the Fourth International Conference on Music Information Retrieval, pp. 193–199.
21. E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. IEEE Multimedia, 3(3):27–36, 1996.