9th International Conference on Music Perception and Cognition

Alma Mater Studiorum University of Bologna, August 22-26 2006

Muugle: A music retrieval experimentation framework

Martijn Bosma, Remco C. Veltkamp, Frans Wiering

Utrecht University

Padualaan 14, 3584 CH Utrecht, The Netherlands

{mbosma, remco.veltkamp, frans.wiering}@cs.uu.nl

ABSTRACT

Recently many music information retrieval (MIR) methods have been developed; most of these employ a music representation and have a notion of musical similarity. However, there is still no user-centered framework that allows the comparison of their different MIR techniques. This paper describes Muugle (Musical Utrecht University Global Lookup Engine), which is such a framework. Currently it has four interfaces for query formulation and a piano-roll editor for query modification. These interfaces and the editor are the subject of a usability study described in this paper. Feature-extraction and matching algorithms are implemented as components which can be selected by the user after query formulation. The features of the query are compared with those of the music stored in the Muugle database. As a first instantiation the database was filled with two music collections, namely 815 ringtones, and 476.621 incipits of the RISM collection. Muugle employs the vantage indexing method to enable fast retrieval results. This work concludes with the discussion of some promising higher-level feature extraction methods currently under development.

Keywords

Music Information Retrieval, Framework

INTRODUCTION

Music information retrieval (MIR) is the interdisciplinary science (combining computer science and musicology) of retrieving musical objects. Recently many MIR methods have been developed. Typke et al. (2005b) provide a survey on the state of the art of MIR systems. It becomes clear from this survey that there are many differences between these systems. For example, the representation of the music data on which they operate varies. Some only use audio data, like Shazam (Wang, 2003), while others, like Themefinder (Kornstadt, 1998), use only symbolic data.

The sizes of the collections on which these systems are tested are very different as well. Meldex (Bainbridge et al., 2004) was for example tested on a database of 9,354 symbolically stored tunes while Shazam operates on a database of more than 2.5 million audio fingerprints. While Shazam only searches for identity, most of these methods have a concept of musical similarity. However, very different notions of musical similarity exist, ranging from the specific similarity of closely-related melodies and themes to the generic similarity of works by the same composer or music in a similar style or genre (Typke et al., 2005b, Figure 1).

In: M. Baroni, A. R. Addessi, R. Caterina, M. Costa (2006) Proceedings of the 9th International Conference on Music Perception & Cognition (ICMPC9), Bologna/Italy, August 22-26 2006.

MIR systems have a different notion of musical surface, the atomic level of music representation within a system (Nattiez, 1975). Details below this level are ignored. Clearly, audio and symbolic MIR systems deal with very different musical surfaces. Other differences include the ability to handle polyphony, and the inclusion of rhythm in the retrieval process.

All these systems thus operate under different circumstances. To make a methodical comparison of their performance possible, the underlying techniques of these systems need to operate in the same framework.

Related Work: A framework where feature extraction methods are implemented as components is RUBATO (Mazzola and Zahorka, 1994). RUBATO is designed to analyze the structure of score data, and to use this analysis to generate performance data. It translates raw notes of the score to a human-like performance. RUBATO is *not* designed for music retrieval purposes, so we do not find any similarity components here. However, RUBATO does have feature extraction methods which are implemented as Analytical Rubettes. These Rubettes can be plugged into the framework, thereby allowing RUBATO to compare the effect of different feature extraction methods on music performance.

MIREX (Music Information Retrieval Evaluation eXchange) is a competition in which MIR systems are compared (http://www.music-ir.org/mirex2006). Researchers participate by sending their MIR system as a Virtual Research Lab in the M2K (Music-to-Knowledge) framework (Downie et al., 2004). This framework enables the researcher to define matching and extraction methods as a set of components and it gives standardized ways to link them in a dataflow network. In the MIREX contest these Virtual Research Labs operate on a common database, which is embedded in the framework. The M2K framework is used in the performance evaluation of MIR systems at the MIREX competition. The focus is not on MIR system users, neither is it designed as a framework for usability experiments. Muugle however does focus on the user and on usability testing.

Contribution: To our knowledge, there is still no user-centered framework that allows the comparison of different feature extraction and similarity methods. Our aim with Muugle (*Musical Utrecht University Global Lookup Engine*) is to provide such a framework, where different MIR methods for feature extraction, matching and presentation can be compared. With all methods operating under the same circumstances, a methodical comparison of their performance is made possible.

This paper gives an overview of the Muugle framework. Its architecture and its components are discussed in the next section. Then a description of the current instantiation (the particular modules plugged into the framework) is given and we conclude the paper with a short discussion of current research activities centered on Muugle.

ARCHITECTURE

A retrieval process starts with the query formulation in the user interface and it leads to the presentation of the most similar music present in the database. Many components and processes are involved to make this possible. Here a brief overview of the system is given.

Figure 1 displays the architecture of the Muugle framework. The arrows in the diagram represent flows, the ellipses processes and the boxes data.

A distinction is made between the preprocessing steps indicated by the light grey area, and those running at query-time. Preprocessing steps are feature extraction of the music stored in the database and index construction. Query-time processes are query formulation, feature extraction from the query, matching and fetching. The user interface components are depicted in the dark grey area.

The upper left corner of Figure 1 shows the music data. This data set can be a local music collection on hard-disk, or music found by a web-crawler on the internet. Feature extraction components extract features from the tunes of this collection and store them in the database's tables. Each matching component may require a specific feature set to be extracted from the data.

Comparing the features of the query with those of all tunes in the database can be a time-consuming task. Therefore an index data structure is constructed. At query time a matching component only has to make a limited number of comparisons to locate matching items. However, not all matching methods are equally suited for indexing.

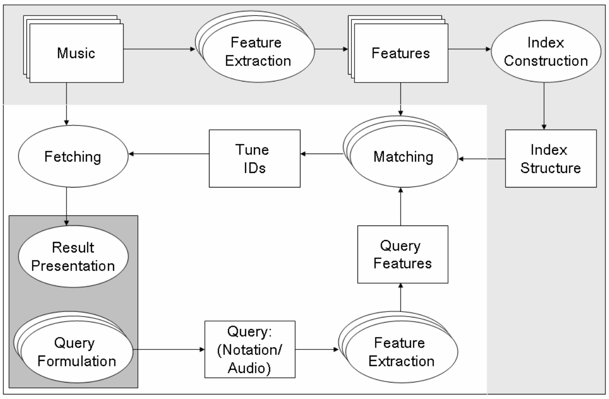

Figure 1 Overview of the Muugle-system architecture

The user can formulate a query in several ways. The query representation can be symbolic (for example MIDI) or audio (for example wav, MP3). Appropriate features must be extracted from it for matching. The user-selected matching component uses the features of the query and those of the tunes in the database to arrive at a similarity judgment.

The output of the matching component is an ordered list of tune-ids. This list consists of references to the tunes stored in the database. It is ordered according to the tunes' similarity, with the most similar ones at the top. The fetching

ISBN 88-7395-155-4

module receives this list of tune references and finds the corresponding data in the music box. Finally these results are presented to the user, who has the possibility of giving relevance feedback.

Query formulation

A query formulation interface can be regarded as a bridge between the MIR system and the outside world. The user on one side of the bridge has notions of the melody to be searched, and motor-control abilities such as the use of arms, hands, and vocal apparatus. On the other side there is the database with music features.

Query formulation can be regarded as crossing the bridge from the user to the system. This crossing consists of the transformation of brain activation to the representation of query features, where intermediate representations might consist of muscle movements, changes in air pressure (sound), or series of events such as mouse clicks.

In Muugle a query formulation interface consists of three functional parts: input, edit and search. Muugle currently provides four of these interfaces, each having a different input part. The edit part consists of a basic piano-roll editor. They are discussed in section “An initial instantiation of Muugle”.

Finally, the search part consists of a matching component drop box and a search button. After selecting the matching component, the search button must be pressed to send the query to the server where the features will be extracted and matched by the selected matching algorithm.

Feature extraction

Different matching algorithms may operate on different features. Many algorithms make use of basic features on the note level, such as onset time, duration, and pitch. Algorithms with more knowledge of music can be constructed which need high-level features such as chords, key or meter. An algorithm can operate on the audio level as well. For example a matching algorithm like Shazam uses audio fingerprints.

Index construction

For simple similarity searches on features, standard indexing methods provided by the database management system, such as B-trees, could be used. For complex similarity measures such as the transportation distances (Typke et al., 2003), this is not possible. However, if the triangle inequality holds for the distance function, other indexing methods can be used to avoid processing all database tunes. One is the vantage indexing method (Vleugels and Veltkamp, 2002; Typke et al., 2004). The idea is to pre-calculate the distances between all database tunes and some vantage objects and store them in a table. The query object has a certain distance to these vantage objects. The set of tunes that have about the same distance to the vantage objects contains also those objects that have about the same distance to the query object. So at query time only the possibly complex calculation between the query and the vantage objects must be made, and then an efficient range search among the pre-computed distances can be done.

This indexing method is generally suitable for any similarity measure that satisfies the triangle inequality.

Matching

Computing the distance between two feature sets to arrive at a dissimilarity measure is the task of the matching components of Muugle. Every interface contains a matching component drop box, from which the user can select the preferred matching component. This selection also determines which features will be extracted from the query.

Because every matching component operates on different features in different ways, we can expect different search results. The comparison of different matching and feature extraction components and the possibilities of relevance feedback lie at the heart of our research.

Presentation

After a query has been formulated, and the similar items have been retrieved, the results are displayed. Usually, result items consist of two parts: metadata in the form of text, and the music itself. The music can be presented visually and acoustically. When presented visually the focus can be on the individual items or on the relation between them. An example of the former is a list such as the result presentation of Themefinder, where the score is shown for every item. An example of the latter is the “Islands of Music” visualization (Pampalk et al., 2004). Here a music collection is visualized as a map, with the music items positioned on locations that correspond to different values of music features. This type of visualization is especially suitable for browsing.

A good presentation of the results depends on the users musical abilities, notions of similarity and retrieval needs. A Shazam user is interested in the metadata of a song while hearing its content. It is clear that the result presentation should simply consist of the metadata of that particular song. On the other hand a Themefinder user is interested in the occurrence themes in musical works, and therefore the score of multiple results together with the metadata is displayed.

INITIAL INSTANTIATION OF MUUGLE

Muugle can be found on *http://give-lab.cs.uu.nl/muugle*. In this first instantiation of Muugle the database gives access to two music collections, namely 815 ringtones, and 476.621 incipits of the RISM collection (RISM, 2002). These collections contain 32.718 and 8.401.975 notes respectively. For each note the onset pitch and duration are stored. From the ringtones and incipits chunks are extracted, each consisting of six consecutive notes in the ringtones and five in the RISM collection. Chunks overlap resulting in 28.483 chunks in the ringtone and 4.664.702 in the RISM collection. These chunks are stored in a feature table of the database and are indexed by means of the vantage indexing method with 15 chunks as vantage objects.

The ringtone collection can be regarded as a toy problem, while the RISM collection puts higher demands on the efficiency of the algorithms. In the latter case an indexing structure as created by the vantage indexing method is absolutely necessary to do any sensible real time retrieval.

Query formulation

Currently the user can choose from four different interfaces to formulate a query, namely the Software Keyboard, the File Upload Menu, the MIDI Device Interface, and the Query by Humming interface. A formulated query can be modified in the piano-roll editor.

Software keyboard

The software Keyboard interface as depicted in Figure 2, allows query formulation in two ways: by clicking with the mouse on piano keys on the screen and by typing letter keys of the computer keyboard. When the piano keys are clicked or letter keys (*a,w,s,e,d,f,t,g,y,h,u,j,k*) are pressed the onset time, pitch and duration of the corresponding notes are recorded and a piano sounds of these notes are heard. The notes will be shown in the piano-roll bar below the piano keys. This piano-roll bar displays the query in the piano-roll notation. It is meant to give the user an overview of the imported notes.

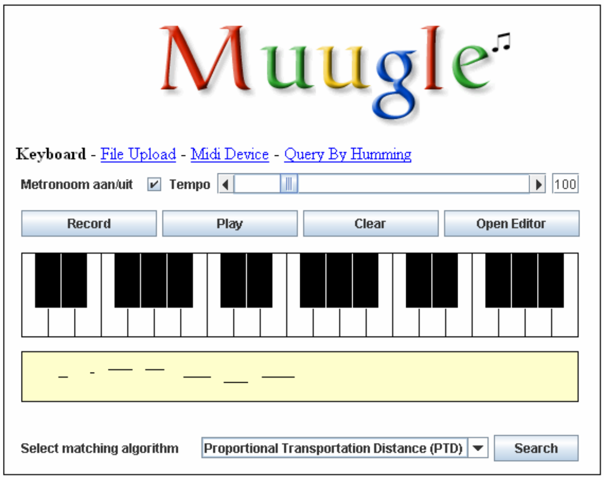

Figure 2 Software Keyboard

File upload menu

In the File Upload Menu a query can be formulated by uploading a MIDI file. A MIDI file can be selected by pressing the Browse button or by typing the file name and path in the corresponding text field.

MIDI device interface

When a MIDI device is connected to the computer, this interface can be used. The recording process is the same as that of the software keyboard, but now the input is originating from the MIDI device.

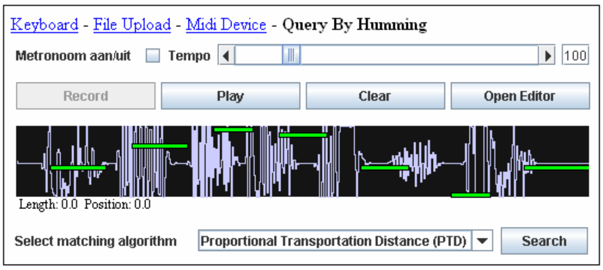

Figure 3 Query by Humming Interface

Query by Humming

Query by Humming is the activity of formulating a query by singing or whistling. This query is recorded and the features needed for MIR are extracted from the audio signal. In our case these features are onset-time, pitch and duration. Muugle employs two alternative methods: one on the basis of the so-called Enhanced Summation Auto-Correlation Function (Tolonen and Karjalainen, 2000), and one used in Musipedia (http://www.musipedia.org/) on the basis of the Fast Fourier Transform, described in Prechelt and Typke 2001. Figure 3 shows the Query by Humming interface. The wave file of the recorded melody is displayed together with the onset, pitch and duration information as found by the pitch finding algorithm. The query can be modified in the piano roll editor.

Piano roll editor

The edit part consists of a basic piano-roll editor (Figure 4), where a query can be adjusted or generated from scratch by mouse-clicking on the notes. Editing operations are the addition, removal, stretching and position-change of the notes.

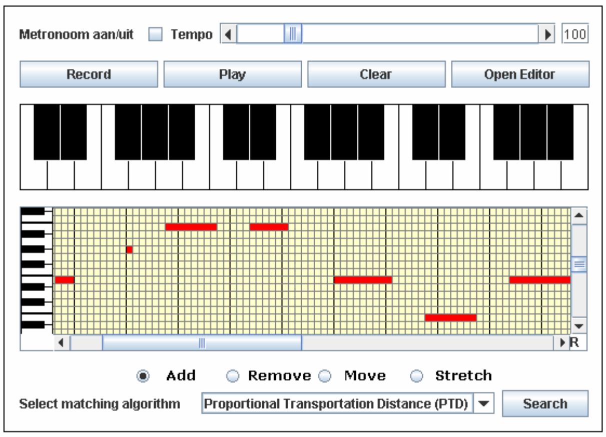

Figure 4 Software Keyboard with unfolded piano-roll editor.

Matching components

Currently there are four matching components implemented in Muugle, which all operate on the note level. They need the onset, pitch and duration features of the notes. Three of these components use transportation distances, which are the *Earth Mover's Distance* (EMD), the *Proportional Transportation Distance* (PTD) (see Typke et al., 2003, 2005 and 2006), and the *Combined EMD-PTD*. The fourth component uses the *Maximum Overlap* algorithm (P3) of Ukkonen et al., 2003 which is one of the geometrical algorithms developed for the C-Brahms project (Lemström et al., 2003).

Every chunk is indexed twice: once with the EMD and once with the PTD as distance measure. Currently there is no index for the maximum overlap algorithm.

The RISM collection can be searched with the PTD component, and the ringtone collection can be searched with all four components.

Earth mover's distance EMD

The EMD compares weighted point sets. As a metaphor one point set can be regarded as heaps of sand and the other as holes in the ground. The EMD computes the minimum amount of work needed to fill the holes with sand. A detailed description of the algorithm can be found in Typke et al., 2003, 2006.

The EMD component of Muugle compares the weighted point sets of chunks instead of whole tunes. For every chunk of the query the tunes containing the most similar chunks in the database are retrieved. Each of these tunes receives a score that corresponds to the highest rank of one of their chunks in the EMD result list. The overall similarity measure of a tune is the median ranking of its chunks. This is the output of the EMD matching component.

Proportional transportation distance: PTD

The PTD matching component is very similar to the EMD component. The only difference is that there is an extra step performed in the generation of the weighted point sets. After generating a weighted point set the weight of every point in the point set is divided by the set's total weight. This means that the total weight of a PTD point set is always 1.

As a result the triangle inequality holds, which is an advantage when using the vantage indexing method. On the other hand partial matching is not possible.

Combined EMD-PTD

This matching component is a combination of the PTD and the EMD. It is based on the idea that when two chunks are considered very similar by one of the two algorithms, there should be something' relevant in the data that causes it, and this should not be ignored. Therefore this matching component computes for every chunk comparison both the EMD and PTD similarity distance. Then the minimum of the two is chosen.

Maximum overlap

The Maximum Overlap algorithm is one of the 3 geometrical algorithms (P3) that were developed by Ukkonen et al., in the C-Brahms project (Ukkonen et al., 2003), (Lemström et al., 2003).

For this algorithm notes are represented as line segments in a 2D plane, just like piano-roll notation. Given a query and a tune of the database, this algorithm computes efficiently the maximum possible overlap of the line segments divided by the total length of all the segments of the tune in the database. There is no chunking involved in this algorithm.

Result presentation

Currently the results are presented as a scrollable list as shown in Figure 5. The results are ordered according to their similarity to the query, with the most similar ones at the top. Every result consists of the score notation of its first few bars. The query is displayed again in the piano-roll bar below the keyboard on the top of the screen. To obtain the score from the MIDI data LilyPond (Nienhuys and Nieuwenhuizen, 2003) was used. LilyPond is an automated engraving system. It formats music notation automatically without requiring typographical expertise of its users.

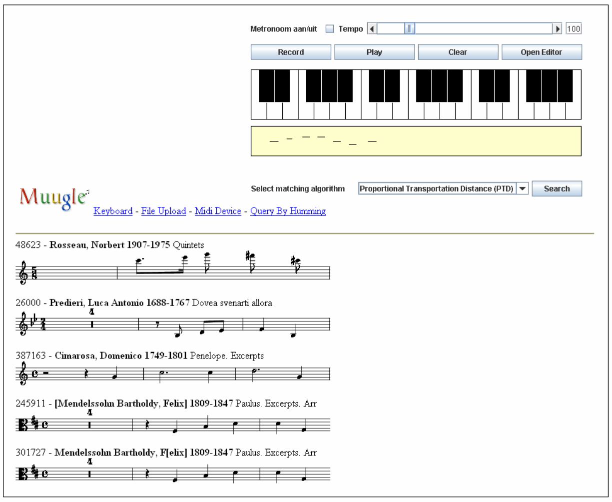

Figure 5 Result List

MUUGLE AS AN EXPERIMENTATION PLATFORM

Muugle is a framework that allows the evaluation of many aspects of MIR. Its modular architecture is designed to facilitate experiments, by comparing different feature extraction and matching components. Furthermore, usability experiments on different query formulation methods can be conducted and different ways of result presentation can be studied and compared by implementing different presentation components.

There were 32 subjects involved in the experiment. Before they started the query formulation task they were asked whether they played piano; whether they played any other instrument and whether they had any experience with MIDI software. We expected that these features could have an impact on the results.

Eight input method conditions were tested:

- Computer keyboard: keys on the computer keyboard
- Software keyboard: piano keys on the screen
- MIDI device: connected Terratec MIDI keyboard
- Piano-roll editor
- Computer keyboard with piano-roll editor
- Software keyboard with piano-roll editor
- External keyboard with piano-roll editor
- Query by Humming

All 32 subjects used all eight input methods in a random predefined order. We selected eight monophonic example melodies from the ringtone collection, with a length ranging from four to nine seconds. For every subject every input method was randomly paired with one of these melodies.

Before query formulation, the subjects were allowed to hear an example melody as often as they thought was necessary to remember it. However, during query formulation, it was impossible to hear it again.

Two experts rated the similarity between the resulting queries and the corresponding example melodies, on a 5 point scale. The experts did not know what the used input method was and which subject formulated which query.

The evaluation of the initial results suggest that the Query by Humming method is slightly better when the sound registration works correctly, and that subjects with prior knowledge of MIDI benefit from the piano-roll editor. A larger experiment is needed to give support to these indications.

DISCUSSION AND FUTURE WORK

The four matching methods that have been implemented so far, all operate on the note level. However, music cognition research has shown that we memorize and compare music generally above the note level. Besides, we do extremely well in recognizing tunes even when they are heavily distorted or varied. Therefore we believe that the performance of a MIR system will improve if it operates on higher-level features that are relevant to human music perception and cognition.

Along this line we are developing a feature extraction module that extracts a key profile from the music. It is based on based on research by Krumhansl, (1990), who developed a key-finding algorithm that uses the results of her experiments on the sense of key in humans. We believe that using this key profile representation for matching gives us interesting results on tonal similarity.

Another feature module under development detects metric structures by finding regular patterns in the note onsets of a piece (Temperley, 2001). It consists of a set of preference rules that describe the constraints of a well-formed beat assignment (Lerdahl and Jackendoff, 1983). For example the algorithm prefers a metrical structure that aligns strong beats with note onsets and it prefers the beats to be evenly spaced in time. We are interested in the relative position of the notes to the beat, which will be used to retrieve music.

Furthermore we have started investigating the presentation of the output and the possibility of relevance-feedback by the user. For this first instantiation of Muugle we have chosen to present the results in the form of a list. This is just one way of presenting the query results. Other presentations such as the “Islands of Music” visualization for example, might be better in a situation where the user is able to browse through the results.

Currently, the Alexa Service is used to build a collection of references to MIDI files available on the web (http://websearch.alexa.com). Relevant features and the URLs of these files are stored in the database. Right now the Alexa web crawlers have gathered in the order of 100.000 tunes, and this collection is still growing. In the end the size of the collection will be much larger than is used so far for experimenting with music retrieval algorithms. This will put much higher demands on the efficiency of the indexing and the matching components.

ACKNOWLEDGEMENTS

This work is partially supported by the Dutch ICES/KIS III bsik project MultimediaN. We thank Kjell Lemström and the other researchers of the C-Brahms project for providing us their implementation of the Maximum Overlap algorithm. Bart van Andel, Niels Gorisse, Robbert Krijger, Bart Kuipers and Rainer Typke implemented Muugle components. Thanks to Cathelijne van der Veen for discussions on result presentations.

REFERENCES

Alexa Websearch. http://websearch.alexa.com

Repertoire International des Sources Musicales RISM. 2002. http://rism.stub.uni.-frankfurt.de

D. Bainbridge, Cunningham S., & Downie J.S. Greenstone as a Music Digital Library Toolkit. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, 42–43, 2004.

Downie J.S., Futrelle J, & Tcheng D. The International Music Information Retrieval Systems Evaluation Laboratory: Governance, Access and Security. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, 9–14, 2004.

Kornstadt A. Themefinder: A Web-based Melodic Search Tool. *Computing in Musicology*, 11:231–236, 1998.

Krumhansl C. L. *Cognitive Foundations of Musical Pitch*, volume 17 of *Oxford Psychology Series*. Oxford University Press, Oxford, 1990.

Lemström K., Mäkinen V., Pienimäki A., Turkia M., & Ukkonen E. The C-BRAHMS Project. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 237–238, 2003.

Lerdahl F. & Jackendoff R. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA, 1983.

Mazzola G. & Zahorka O. The Rubato Performance Workstation on Nextstep. In *Proceedings of the International Computer Music Association. (ICMC'94)*, San Francisco, CA, 1994.

Nattiez J.-J. *Fondements d'une semiologie de la musique.* Union Generales d'Editions, Paris, 1975.

Nienhuys H.W. & Nieuwenhuizen J. Lilypond, a System for Automated Music Engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, 2003.

Pampalk E., Dixon S., & Widmer G. Exploring Music Collections by Browsing Different Views. *Computer Music Journal, 28(3),* 49–62, 2004.

Prechelt, L. & Typke, R. An Interface for Melody Input. ACM Transactions on Computer-Human Interaction 8(2), 133-149, 2001.

Temperley D. *The Cognition of Basic Musical Structures*. MIT Press, Cambridge (MA) USA, London England, 2001.

Tolonen T. & Karjalainen M. A Computationally Efficient Multipitch Analysis Model," IEEE Trans. Speech Audio Proc. 2000 vol. 8(6), 708-716, 2000.

Typke R., Giannopoulos P., Veltkamp R. C., Wiering F., & van Oostrum R. Using Transportation Distances for Measuring Melodic Similarity. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*,,107–114, 2003.

Typke R., Wiering F., & Veltkamp R. C. A Search Method for Notated Polyphonic Music with Pitch and Tempo Fluctuations. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, 281–288, 2004.

Typke R., Wiering F., & Veltkamp R. C. Evaluating the Earth Mover's Distance for Measuring Symbolic Melodic Similarity. In *MIREX-ISMIR 2005: 6th International Conference on Music Information Retrieval.*, London, UK, 2005a.

Typke R., Wiering F., & Veltkamp R.C. A Survey of Music Information Retrieval Systems. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, London (UK), 153–160, 2005b.

Typke R., Wiering F. & Veltkamp R.C. Transportation Distances and Human Perception of Melodic Similarity. In *Musicae Scientiae,* 2006.

Ukkonen E., Lemström K., & Mäkinen V. Geometric Algorithms for Transposition Invariant Content-based Music Retrieval. In *Proceedings of the 4rth International Conference on Music Information Retrieval (ISMIR 2003)*, 193–199, 2003.

Vleugels J. & Veltkamp R. C. Efficient Image Retrieval through Vantage Objects. *Pattern Recognition*, 35(1):69–80, 2002.

Wang A.. An Industrial Strength Audio Search Algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.