EFFICIENTLY COMPUTABLE SIMILARITY MEASURES FOR QUERY BY TAPPING SYSTEMS

Gunnar Eisenberg, Jan-Mark Batke, and Thomas Sikora

Communication Systems Group
Technical University of Berlin
{eisenberg|batke|sikora}@nue.tu-berlin.de

ABSTRACT

A Query by Tapping system is a database which contains metadata descriptions of songs. The database can be scanned by tapping the melody line's rhythm of a song requested on a MIDI keyboard or an e-drum. For the processing of queries the system computes the similarity of the query and the content inside the database by applying a similarity measure. Due to the high number of comparison processes in large databases efficiently computable similarity measures are needed. This paper presents two efficiently computable similarity measures which evaluate rhythmic properties of monophonic melodies represented in an MPEG-7 compliant manner. The usage and effectiveness is presented and evaluated with the real time capable Query by Tapping system BeatBank.

1. INTRODUCTION

Well known Music Information Retrieval (MIR) systems in the context of meta-information processing are Query by Humming (QBH) systems. QBH systems are used to search songs by humming their melody into a microphone [1].

Less known MIR systems are Query by Tapping (QBT) systems which allow users to formulate a query by tapping the rhythm of the song's melody. In QBT-systems pitch information is normally not taken into account in any way. The tapping of the rhythm is performed on a MIDI keyboard or on an e-drum. Some systems even allow users to tap the melody's rhythm as an acoustic process like clapping or knocking which is recorded and processed [2].

An integral component of every QBH or QBT system is the similarity measure which is an algorithm for the computation of the similarity of two database entries. This paper focuses on algorithms for the comparison of melody's rhythms presented in an MPEG-7 compliant manner. These algorithms are primarily needed for QBT systems since they deal with rhythmic properties only. Two efficiently computable similarity measures are presented and discussed in detail in the following. Due to their efficiency they can be applied in a real-time capable QBT system.

The QBT system BeatBank [3] which is used for evaluation purposes operates in real-time and online. This means that after every tap made by the user, the system presents the search's actual result list. The content of the database is saved in MPEG-7 XML documents.

2. PREVIOUS APPROACHES

Theoretical aspects regarding the comparison of two symbol strings have been discussed in several publications. These theories can be applied to the comparison of two rhythms represented in an MPEG-7 compliant manner. A very common method used to measure the similarity of two strings is the so-called Approximate String Matching method [4] which is an application of Dynamic Programming [5]. In addition to these theoretical approaches there are publications which present implementations of QBT systems.

McNab, Smith, and Lloyd carried out experiments on the similarity of melodies using a large database of 9600 songs [7]. One of their goals was to find out which musical errors occur when persons are singing or humming melodies well known to them. They report that test persons generally tend to fill in extra notes or to drop notes when reproducing melodies. Their results can be generalized to the reproduction of rhythms. It is shown that simpler similarity measures lead to longer queries. Consequently, a similarity measure, which only takes rhythmic properties into account, needs longer search queries than a more complex similarity measure. Therefore the similarity measure should be efficiently computable.

Uitdenbogerd and Zobel presented different matching strategies. Some of the experiments were performed with automatically generated queries [8], whereas others are performed with manually generated queries [9]. The manually generated queries were played on a MIDI keyboard and recorded by a MIDI sequencer. Their experimental setup is similar to the setup presented in this paper. One of their main results is that similarity measures which perform well with automatically generated queries do not necessarily yield good results with manually entered queries.

Kim, Chai, and Garcia carried out experiments with different melody representations [1]. Besides the time signature the beat vector was taken into account as a rhythmic property. The proposed comparison process computes the similarity for every single beat. The experiments carried out used automatically generated random queries from the database. It could be shown that the usage of additional rhythmic information allows shorter queries for search processes.

Chen and Chen [10] presented a string matching method which divides rhythm-strings into smaller sub-strings. With their method a tree structure is generated which is searched in greater detail. They present different matching strategies and different definitions of similarity measures.

Jang, Lee, and Yeh presented a QBT system which allows a user to clap or tap the rhythm of the melody requested and record it with a microphone [2]. Queries are processed by an offline process in which the system extracts the notes' durations. The similarity measure which is applied is based on Dynamic Programming [5].

Due to the fact that the presented algorithms use different test scenarios and different databases it is hard to compare their effi-

ciency in terms of computational effort as well as their ability to find similar patterns.

3. THE BEATBANK SYSTEM

3.1. Overview

The BeatBank system is a QBT system which is implemented as a *Virtual Studio Technology* plug-in instrument (VSTi)[1]. With an appropriate VST host, the system operates in real-time and online. This means that a new search result list is computed and presented by the system after every note entered. BeatBank is a free system which could help to define a uniform test scenario at least for the comparison of rhythms. The latest version for Windows as well as the used database can be downloaded for free at our department's website[2].

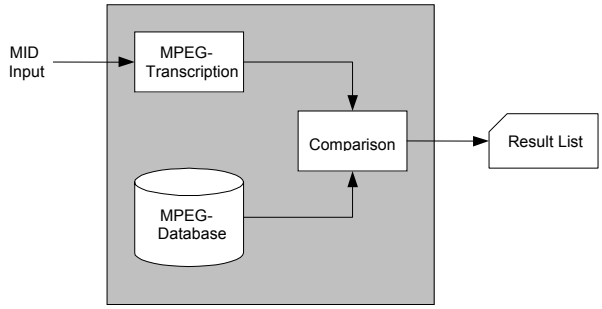

Figure 1: *Flowchart of the System. The MIDI input gets transformed into an MPEG-7 compliant representation. Then a comparison with the database content is performed. The results are presented to the user.*

The user interfaces the system by a MIDI-input device like an e-drum or a MIDI keyboard. While the input query is played, the taps can be acoustically monitored by loudspeakers. The search process' results are presented continuously i.e. the result list is updated automatically after every note played by the user. The entered query gets transcribed into an MPEG-7 compliant representation and compared with the content of the database (see Figure 1). The comparison process uses the similarity measures which are described in Section 4. The rhythms inside the database originate from MPEG-7 XML files which are uploaded into the memory during the system's initialisation.

3.2. MPEG-7 Description of Rhythms

The database content is represented in an MPEG-7 compliant manner, by using the Description Scheme (DS) `MelodyContour`. `MelodyContour` can be used for a loose description of monophonic melodies. It contains the Descriptors `Contour` and `Beat`.

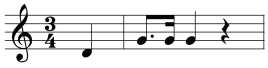

Figure 2: *First bars of the song "O Tannenbaum" which would be represented by a* `Beat` *vector of [3 4 4 5].*

[1] `www.steinberg.com`
[2] `www.nue.tu-berlin.de/wer/eisenberg/beatbank.html`

The Descriptor `Contour` describes pitch information by a five-level contour and is not evaluated by the BeatBank system. The Descriptor `Beat` contains a vector of integers, describing the melody's rhythm. It is formed by numbering every note with the integer number of the last full beat. The beats are being counted continuously, starting with the first beat in the first bar of the melody (see Figure 2). If the first bar is an upbeat the vector's first entry carries implicit information on the length of the upbeat. This results from the beats which passed before the first note occurs in the upbeat. More Information on MPEG-7 can be obtained from [11] and [12].

4. SIMILARITY MEASURES

4.1. Overview

A similarity measure represents the similarity of two rhythms as a decimal number between 0 and 1 with 1 meaning identity.

When comparing two MPEG-7 `Beat` vectors the goal is to find pairs of elements for matching notes. This can be achieved by means of Dynamic Programming and Dynamic Time Warping. However, a computation by these methods (e.g. the Dot-Plot) can be costly and not applicable for real-time capable system [6]. Therefore efficiently computable similarity measures are needed. Two efficiently computable similarity measures named *Direct Measure* and *Wring Measure* are presented in the following. Both utilize certain limitations of the MPEG-7 `Beat` representation.

4.2. Direct Measure

All elements of MPEG-7 `Beat` vectors are positive integers and every element is equal or bigger than its predecessor. These limitations enable a simplified computation of matching elements. This leads to the Direct Measure which is robust against single note failures. For two Vectors U and V it can be computed by the following iterative process:

- Compare the two vector elements u_i and v_j, (starting with i=j=1)
- If $u_i = v_j$ the comparison is considered a *match*. Increment the indices i and j and proceed the comparison.
- If $u_i \neq v_j$ the comparison is considered a *miss*. Increment only the index of the vector whose element has been smaller for the next comparison.
- Continue the comparison until the last element of one of the vectors has been detected as a match or the last element in both vectors is reached.

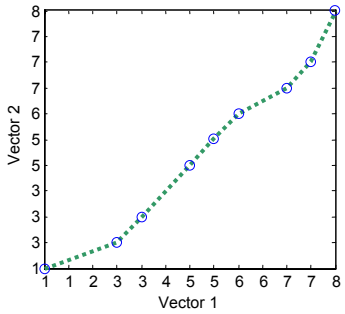

Figure 3: *Comparison path for two vectors compared by the Direct Measure. The circles mark matching elements.*

An example for the pair matching is shown in Figure 3. The similarity A is then computed as the following ratio with T being the number of matches and V being the number of comparisons:

$$A = T / V \tag{1}$$

The maximum number of iterations for two vectors of length N and length M is equal to the sum of the lengths (N+M). This is highly more efficient than a computation with classic methods like the Dot-Plot which needs at least N·M operations [6].

4.3. Wring Measure

With the usage of MIR systems and especially QBT systems several predominant errors occur while users tap queries [3] [7]. Besides single note failures, especially unskilled users sometimes loose their measure and stop tapping. They restart tapping at the start of one of the succeeding bars. The test persons sometimes start tapping one or more bars too soon or too late. The *Wring Measure* tries to compensate these effects by focusing on the evaluation of transitions from one integer value to the next bigger one in the `Beat` vectors. For two vectors compliant to the MPEG-7 `Beat` Descriptor the Wring Measure can be computed by the following three step process:

- Transform both `Beat` vectors $\underline{V}$ into auxiliary vectors $\underline{V}'$. The elements of $\underline{V}'$ are 1 if the corresponding element in $\underline{V}$ is bigger than its predecessor and 0 if it is smaller. The first element of the auxiliary vector is set to 1 by default.
- Build two new `Beat` vectors $\underline{V}^*$ from both auxiliary vectors $\underline{V}'$ by accumulating all values of the auxiliary vectors' elements whose index is less than or equal to the index of the element to be computed.
- Compare the two new `Beat` vectors $\underline{V}^*$ with the Direct Measure.

The following equations give an example for the transformation of a `Beat` vector $\underline{V}$ to a new `Beat` vector $\underline{V}^*$.

$$\underline{V} = [1 \;\; 3 \;\; 3 \;\; 3 \;\; 5 \;\; 5 \;\; 6 \;\; 7 \;\; 7 \;\; 7 \;\; 8] \tag{2}$$

$$\underline{V}' = [1 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 1] \tag{3}$$

$$\underline{V}^* = [1 \;\; 2 \;\; 2 \;\; 2 \;\; 3 \;\; 3 \;\; 4 \;\; 5 \;\; 5 \;\; 5 \;\; 6] \tag{4}$$

The result of a complete comparison process using the Wring Measure can be seen in Figure 4. In this figure the same vectors as in Figure 3 are compared. Vector 2 is transformed by the Wring Measure as shown in equations (2) to (4).

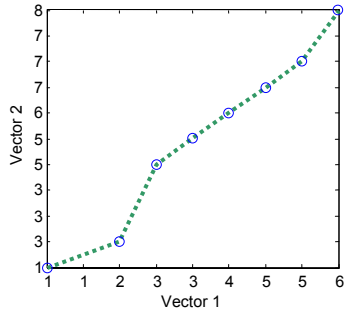

Figure 4: *Comparison path for two vectors compared by the Wring Measure. The circles mark matching elements.*

5. EVALUATION

5.1. Setup

Two major experiments were carried out with the BeatBank system whose database contained 56 pop songs. The songs which were used are the same 47 songs as used by Kim, Chai, and Garcia [1] for their experiments[1] plus nine pop songs which have already been used for prior experiments [3]. These nine songs were used to formulate queries.

The first major experiment was performed with musicians as test persons to test the similarity measures under real world conditions. Four musicians tried to tap the first four bars of each of the nine additional melodies. The persons had to listen twice to the melodies played in cycle mode. Then they had to tap the rhythm of the melody on a MIDI e-drum. The experiment had been repeated four times with different tapping devices: one hand, two hands, one drum stick, two drumsticks. Thus a total number of 144 queries was recorded and evaluated with both similarity measures.

The second major experiment was performed to test the robustness of the two similarity measures against the two sorts of errors which predominantly occur while unskilled users tap queries (loss of measure and early/late start). Therefore the 144 recorded queries were manually transformed into new queries which contain both errors described. The first two recorded bars were delayed by one bar, the second two bars were delayed by two bars thus producing a gap of one bar between the two pieces (Figure 5). This transformation injects errors which mess up the queries quite heavily. The 144 transformed queries were also evaluated with both similarity measures. With both experiments summed up both similarity measures were tested with 288 queries.

5.2. Results

The first major experiment shows that when the queries are evaluated with the Direct Measure 66,7% of the queries determined the correct melody as a best match (see Figure 6). For only 15,3% the requested song was listed worse than the 10th place, which is a very good result. This shows that the Direct Measure is very robust against single note failures.

When evaluated with the Wring Measure 46,5% of the queries determined the correct melody as a best match. For 29,2% the requested song was listed worse than the 10th place. Due to the amount of blur added by the Wring Measure, some correctly played melodies are evaluated wrong. This results in the worse matching outcome compared to the Direct Measure.

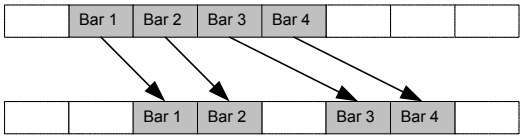

Figure 5: *Transformation of a query to reproduce the two typical sorts of errors which occur while formulating queries (loss of measure and early/late start).*

[1] `www.media.mit.edu/~chaiwei/mpeg7/midi.zip`

The results of the second experiment differ quite much from the results of the first experiment. When evaluated with the Direct Measure 17,4% of the queries determined the correct melody as a best match (see Figure 6). For 62,5% the requested song was listed worse than the 10th place, which is a quite poor result. The Direct Measure is very robust against single note failures but it is vulnerable to the sorts of errors which had been manually inserted for the experiment (loss of measure and early/late start). This leads to the poor matching results.

When evaluated with the Wring Measure 38,2% of the queries determined the correct melody as a best match. For only 32,6% the requested song was listed worse than the 10th place. This is a quite good result with regards to the heavy error injection which had been performed for the experiment. The Wring Measure is more robust against the inserted errors than the direct Measure.

During the experiments it could be seen that single note failures frequently occur in several situations. Users tend to drum along and try to enhance the rhythm with additional strokes, being only poorly similar with the original. When melodies contain long sustained notes users tend to reproduce these by tapping more than just one stroke. This happened especially when they were allowed to use both hands or two sticks for the tapping. People tap more accurately with their hands because they tend to let the drumstick bounce onto the pads of the e-drum.

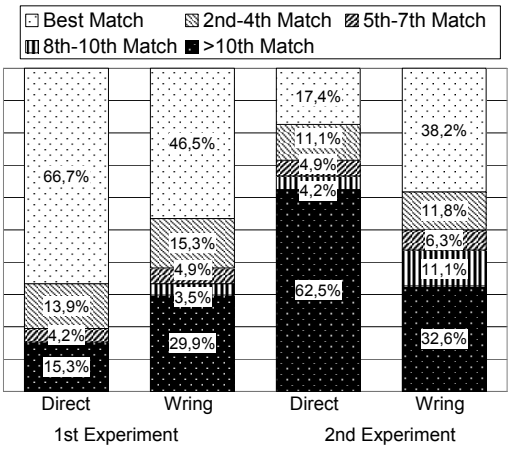

Figure 6: *Mean search results of the BeatBank system for the Direct Measure and the WringMeasure for both major experiments.*

The experiments carried out show that rhythm can be a powerful feature for distinguishing melodies. The introduced efficiently computable similarity measures Direct Measure and Wring Measure yield good results for comparing MPEG-7 compliant rhythms. Each measure has its strength and weaknesses but together they produce good results with linear computational effort. They can be directly used for QBT systems but also help reducing the search duration in other MIR systems. For example, they can be used in QBH systems for the fast setup of reasonable data subsets for further comparison processes with more complex similarity measures.

6. FUTURE WORK

The two presented similarity measures both produce good results in a certain scenario. When used in parallel they produce good overall search results. One probably nonlinear similarity measure needs to be developed that combines the strengths of both measures.

The limitations of rhythm as a feature for distinguishing melodies need to be further investigated with a larger database. Two melodies with a similar rhythm will always be both determined as good match candidates when one of them is played as a query. The mean query's length which is needed for the distinction needs to be determined. This could be done in experiments like those presented by Kim, Chai, and Garcia [1]. It will affect in which situations the melody's rhythm can be used as a single powerful feature.

One of the next versions of BeatBank will have a simple open Application Programmers Interface (API) so that it can use third party similarity measure algorithms compiled into a dynamic link library (dll). This should help to define a uniform test scenario and allows other research groups to test their own similarity measures.

7. REFERENCES

[1] Y. E. Kim, W. Chai, R. Garcia, "Analysis of a contour-based representation for melody," *Proc. Int. Symp. on Music Information Retrieval*, 2000.

[2] J.-S. R. Jang, H.-R. Lee, C.-H. Yeh, "Query by Tapping: A New Paradigm for Content-Based Music Retrieval from Acoustic Input," *Proc. of the Second IEEE Pacific Rim Conf. on Multimedia*, 2001.

[3] G. Eisenberg, J.-M. Batke, T. Sikora, "BeatBank – An MPEG-7 Compliant Query by Tapping System," *Proc. of the 116th AES Conv.*

[4] R. A. Baeza-Yates, C. H. Perleberg, "Fast and practical approximate string matching," *Combinatorial Pattern Matching, Lecture Notes in Computer Science*, vol. 644, pp. 185–192. Springer, 1992.

[5] S. E. Dreyfus, A. M. Law, *The art and theory of dynamic programming*. New York, USA : Academic Press, 1977.

[6] M. Gerstein, *Sequence Comparison via Dynamic Programming*. Tutorial, Yale University, 1998. `http://bioinfo.mbb.yale.edu/course/classes/c3.pdf`

[7] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, S. J. Cunningham, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input," *Proc. ACM Digital Libraries*, 1996.

[8] A. L. Uitdenbogerd, J. Zobel, "Matching techniques for large music databases," *Proc. of the Seventh ACM Int. Multimedia Conf.*, 1999.

[9] A. L. Uitdenbogerd, J. Zobel, "Music ranking techniques evaluated," *Twenty-Fifth Australasian Computer Science Conf.*, 2002.

[10] J. C. C. Chen, A. L. P. Chen, "Query by Rhythm – An Approach for Song Retrieval in Music Databases," *Proc. of the 18th Int. Workshop on Res. Issues in Data Engineering*, 1998.

[11] ISO/IEC JTC 1/SC 29: *Information Technology - Multimedia Content Description Interface*. ISO/IEC FDIS 15938, 2002.

[12] B. S. Manjunath, P. Salembier, T. Sikora, (eds.) *Introduction to MPEG-7 - Multimedia Content Description Interface*. New York, USA: John Wiley & Sons, 2002.