Audio Engineering Society

Convention Paper 6136

Presented at the 116th Convention
2004 May 8–11 Berlin, Germany

BeatBank – An MPEG-7 compliant Query by Tapping System

Gunnar Eisenberg, Jan-Mark Batke, and Thomas Sikora

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

ABSTRACT

A Query by Tapping System is a multi-media database containing rhythmic metadata descriptions of songs. This paper presents a Query by Tapping system called BeatBank. The system allows to formulate queries by tapping the melody line's rhythm of a song requested on a MIDI keyboard or an e-drum. The query entered is converted into an MPEG-7 compliant representation. The actual search process takes only rhythmic aspects of the melodies into account by comparing the values of the MPEG-7 Beat Description Scheme. An efficiently computable similarity measure is presented which enables the comparison of two database entries. This system works in real-time and computes the search process online. It computes and presents a new search result list after every tap made by the user.

1. INTRODUCTION

Descriptive meta-information is needed to differentiate the content in multi-media databases reasonably. This information usually contains both high-level content descriptions and low-level signal descriptions.

The standard MPEG-7 (ISO/IEC 15938) [5] is a multimedia description language which allows multi-media data to be described in different ways. Based on this meta-information it is possible to search in databases by formulating content related queries.

A typical *Music Information Retrieval* system in this context is a *Query by Humming* system which is used to search a song by humming its melody into a microphone [2] [10].

This paper is devoted to a similar topic and presents a *Query by Tapping* system which allows users to formulate a query by tapping the rhythm of the song's melody. Pitch information is not taken into account in any way. The tapping of the rhythm is performed on a MIDI keyboard or on an e-drum. The system operates in realtime and online which means that after every tap made by the user, the system presents the actual search result list. The content of the database is saved in MPEG-7 XML documents.

An integral component of the system is the algorithm for the efficient computation of the similarity of two rhythms represented in an MPEG-7 compliant manner. This algorithm will be discussed in detail in the following.

2. PREVIOUS APPROACHES

Several publications discuss theoretical aspects regarding the comparison of two symbol strings. A very common method used to measure the similarity of two strings is the so-called *Approximate String Matching* method [1] which is an application of *Dynamic Programming* [3] [15]. In addition to these theoretical approaches there are several publications which present implementations of Music Information Retrieval systems.

Sankoff and Mongeau performed a study on the comparison of melodies which is described in [13]. In their Paper differences in rhythm are represented by subtracting the notes' lengths. The similarity of the notes is expressed as a linear combination of the similarities of the pitch and the length of the notes. The weights of the addends are achieved in a heuristic manner. The results of this paper can also be applied to systems using rhythmic aspects only.

McNab, Smith, and Lloyd carried out experiments using a large database of 9600 songs [11] which basically followed the scheme presented by Sankoff and Mongeau in [13]. One of their main goals was to find out which musical errors occur when persons are singing or humming melodies well known to them. They describe that test persons generally tend to fill in extra notes or to drop notes when reproducing melodies. These results can be generalized to the reproduction of rhythms. This shows that a robust similarity measure should not put too much weight on single note failures. It is shown that the number of notes needed for a successful song search grows logarithmically with data bank size and that long queries allow the usage of simpler similarity measures. 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 [14] [16]. They proposed a three-step process (melody extraction, normalization, comparison). In their papers comparison strategies are divided into Dynamic Programming processes and the N-Gram method. Some of the experiments are performed with automatically generated queries [14], whereas others are performed with manually generated queries [16]. The manually generated queries are played on a MIDI keyboard and recorded by a MIDI sequencer. The experimental setup is similar to the system presented in this paper. The experiments show 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 [9]. The time signature as well as the beat vector were taken into account as rhythmic properties. The comparison process they propose computes a similarity for every single beat. A validating experiment was also carried out, which used automatically generated random queries from the database. It was shown that the use of additional rhythmic information allows shorter queries for search processes.

Paulus and Klapuri compared arbitrary rhythms from audio signals [12]. They present an algorithm which compares generic datasets. The presented similarity measure is based on Dynamic Programming. Although generic data sets are compared the algorithm can be adapted to the comparison of datasets which are notated in an MPEG-7 compliant manner.

Ghias, Logan, and Chamberlin describe a Query by Humming system which contains a database of 183 songs in MIDI format [4]. In their system the query can concern any part of the song. The comparison is an implementation of the Approximate String Matching method. The comparison searches for substrings while a fixed number of single errors must not be exceeded. The limiting number of errors can be user-defined or determined heuristically with respect to the query length.

Jang, Lee, and Yeh presented a Query by Tapping system which allows the user to clap or tap the rhythm of the melody requested and record it with a microphone [7]. The system performs an offline process in which it extracts the duration of the notes and searches with a similarity measure based on Dynamic Programming [3] [15]. The experiments carried out demonstrate that the beat information is an effective feature for the song search in a large music database and helps to achieve a satisfactory recognition rate. The same authors also described a Query by Tapping system [8] in the context of the Music Information Retrieval system *Super MBox* which is introduced in [6].

3. SYSTEM DESCRIPTION

3.1. Overview

The BeatBank system is implemented as a *Virtual Studio Technology* plug-in instrument (VSTi). The VST technology has been developed by Steinberg[1]. With an appropriate host, the system operates in real-time and online. This means the system computes and presents a new search result list after every entering of a note. The latest version of BeatBank for Windows can be downloaded for free at our department's website[2].

3.2. User Interface

The user interface of the system consists of 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. Figure 1 shows the setup of the system.

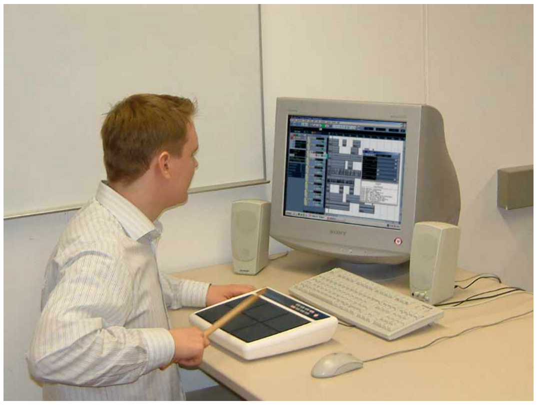
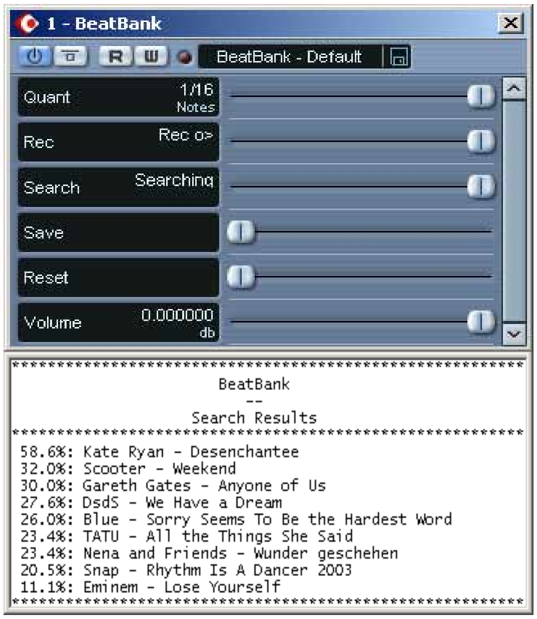


Figure 1: The setup of the BeatBank system consisting of a MIDI e-drum and the computer hosting the plug-in.

The interface window of the system is presented in Figure 2. The upper part of the window allows the user to set all necessary parameters such as the quantization (Quant) or the volume (Volume). This window contains the switches which start the recording of a query (Rec) and the search process itself (Search). The recorded rhythm can be saved as an MPEG-7 XML file on the database (Save).

The search process' results are presented continuously. The result list, which is shown in the lower part of the window, updates automatically after every note played by the user.



Quant 1/16 Notes
Rec Rec o>
Search Searching
Save
Reset
Volume 0.000000 db

```
*******************************************************
                      BeatBank
                         --
                   Search Results
*******************************************************
58.6%: Kate Ryan - Desenchantee
32.0%: Scooter - Weekend
30.0%: Gareth Gates - Anyone of Us
27.6%: DsdS - We Have a Dream
26.0%: Blue - Sorry Seems To Be the Hardest Word
23.4%: TATU - All the Things She Said
23.4%: Nena and Friends - Wunder geschehen
20.5%: Snap - Rhythm Is A Dancer 2003
11.1%: Eminem - Lose Yourself
*******************************************************
```

Figure 2: Interface window of the BeatBank plug-in (version 1.0). All necessary parameter settings are made in the upper part. The lower part presents the search results with the similarity values at the left side.

3.3. Internal Setup

The BeatBank plug-in is hosted by a VST capable host, managing the general MIDI and audio communication. The system internally transcribes the user query into an MPEG-7 compliant representation and compares it with the content of the database, by using a similarity measures which is described in section 5 (see Figure 3).

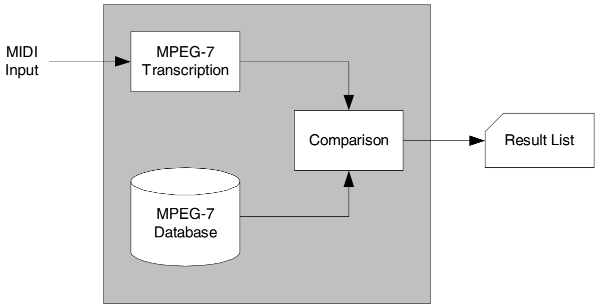


Figure 3: Flowchart of the System.

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

The entries of the database consist of MPEG-7 XML files. To enhance the performance, the content of the XML files is uploaded into the memory during the initialisation of the system.

4. INTERNAL DESCRIPTION OF RHYTHMS

4.1. MPEG-7 MelodyContour

The database content is represented in an MPEG-7 compliant manner, by using the Description Scheme (DS) `MelodyContour`. `MelodyContour` which is shown in Figure 4 can be used for a loose description of monophonic melodies. It contains the Descriptors `Contour` and `Beat`.

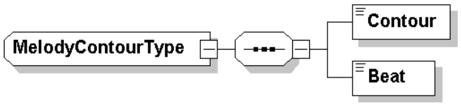


Figure 4: Graphical definition of the MPEG-7 Description Scheme `MelodyContour`.

The Descriptor `Contour` describes pitch information by a five-level contour and is not evaluated by the system. The Descriptor `Beat` represents the actual rhythmic information.

4.2. MPEG-7 Beat

The Descriptor `Beat` contains a vector of integers, describing the melody's rhythm. The vector 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 note of the melody. The beats during the pause of an upbeat are also counted. Thus the first entry of the vector carries implicit information on the length of the upbeat.

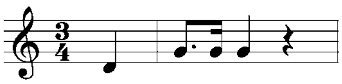


Figure 5: First bars of the song "O Tannenbaum" which would be represented by a `Beat` vector of [3 4 4 5].

More Information on MPEG-7 can be obtained from [5] and [10].

5. SIMILARITY MEASURE

5.1. Overview

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

Two vectors of different lengths need to be compared to compute the similarity of two rhythms represented in MPEG-7. The goal is two 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 can be costly.

5.2. Dynamic Programming

The comparison of two vectors of differing lengths entails different problems, as every element of the first vector can be matched onto every element of the second vector with a certain probability. Thus an algorithm, which matches single elements, needs to be applied to compare two vectors of different sizes.

Theoretically, this matching process can be performed for two vectors V and W with different lengths

$$\underline{V} = [V_1 \quad V_2 \quad \cdots \quad V_N] \tag{1}$$

$$\underline{W} = [W_1 \quad W_2 \quad \cdots \quad W_M] \tag{2}$$

using a matrix P as shown by equation (3) which contains the probability of every element of the first vector to match up with every element of the second vector.

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,N} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M,1} & P_{M,2} & \cdots & P_{M,N} \end{bmatrix} \tag{3}$$

An element $P_{j,k}$ of the matrix gives the probability that the two elements V_k and W_j match up.

The computation of the matrix P from equation (3) would be a recursive process which considers conditional probabilities of neighbor elements. To compute the similarity of the two vectors it is necessary to build a path through the matrix P which connects high probability values. This path also connects the according elements. A weight G is dedicated to each possible step that can be taken through the matrix. The actual similarity measure can be computed by taking the mean value of the single probabilities $P_{j,k}$ of the matrix P weighted with weights G.

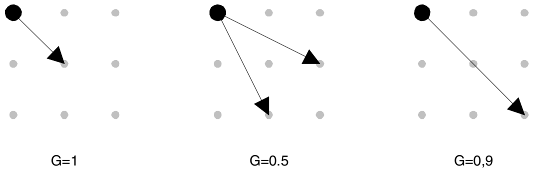


Figure 6: An example of different step patterns that can be taken through the matrix P and the appropriate weights G.

There are several algorithms [3] [12] which are not discussed thoroughly in this paper. The reader has to keep in mind that the calculation of these algorithms can be computationally quite costly. For two vectors of length N and M, the operations needed to build the matrix P from equation (3) are at least the product of the lengths (N·M).

5.3. Direct Measure

The `Beat` vectors which represent rhythms in MPEG-7 have certain limitations allowing an efficient computation of similarity measures. All elements of the vectors are positive integers and every element is equal or bigger than its predecessor. Due to this limitations, it is possible to perform a simplified computation of the path through the matrix P from equation (3) and to find matching elements. This leads to efficiently computable similarity measures such as the *Direct Measure*.

For two vectors compliant to the MPEG-7 `Beat` Descriptor the Direct Measure can be computed by the following iterative process:

- Compare the two elements.
- If they are equal, mark them as a matching pair and compare the next two elements. This comparison step is considered a match.
- If they are not equal, the next element from the vector whose element has been smaller will be taken for the next comparison. This comparison is considered a miss.
- Continue the comparison until the last element of one of the vectors has been connected as a match or the last element in both vectors is reached.

The similarity A can be computed as the following ratio with T being the number of matches and V being the number of comparisons.

$$A = \frac{T}{V} \tag{4}$$

The maximum number of iterations for two vectors of length N and length M is equal to the sum of the lengths (N+M), whereas the maximum number of operations for the matrix P from equation (3) is equal to the product of the lengths (N·M).

This shows that the Direct Measure can reduce the overall operations for the computation of the similarity dramatically. This is even more important since the search in a large database leads to a huge number of comparison processes. The use of BeatBank shows that the vectors to be compared are often longer than 500 entries.

To give an example the following vectors have been compared in Figure 7:

$$\underline{V_1} = \begin{bmatrix} 1 & 1 & 2 & 3 & 3 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 8 \end{bmatrix} \tag{5}$$

$$\underline{V_2} = \begin{bmatrix} 1 & 3 & 3 & 3 & 5 & 5 & 6 & 7 & 7 & 7 & 8 \end{bmatrix} \tag{6}$$

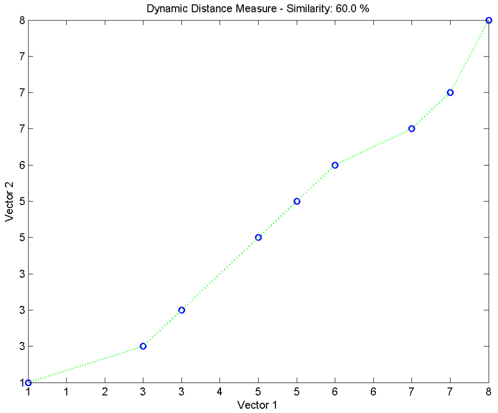


Figure 7: Comparison path of two vectors compared by the Direct Measure. The circles mark matching elements.

6. VALIDATION

The test performed had been carried out with a database containing nine pop songs (see Table 1). Three musicians tried to tap the first four bars of each of the nine melodies and thus produced 27 test-queries.

1.	TATU – All the Things She Said
2.	Scooter – Weekend
3.	Kate Ryan – Desenchantee
4.	Blue – Sorry Seems To Be the Hardest Word
5.	Gareth Gates – Anyone of Us
6.	DsdS – We Have a Dream
7.	Eminem – Lose Yourself
8.	Nena and Friends – Wunder geschehen
9.	Snap – Rhythm Is A Dancer 2003

Table 1: Songs in the database.

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 drum stick, two drumsticks, one hand, two hands.

The results of the experiment show that 76.6% of the queries found the correct melody as a best match (see Figure 8). 16.0% of the queries found the requested song as the second hit in the list. 7.4% of the queries listed the requested song as the third place or worse. This result could be expected due to the special experimental setup and the small database size. Moreover, the experiment has an almost deterministic character besides playing failures of the test persons.

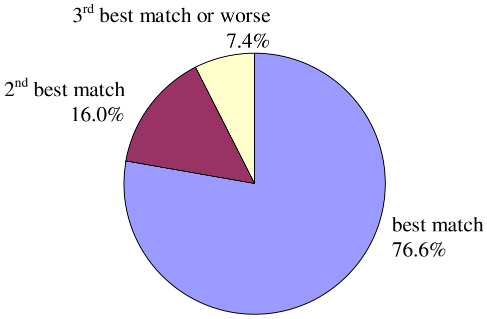


Figure 8: Mean search results of the BeatBank system for the example queries.

To get an idea of how difficult it is to tap the nine rhythms see Figure 9 showing the mean similarity rate of the specific search queries with the target songs averaged by the three players.

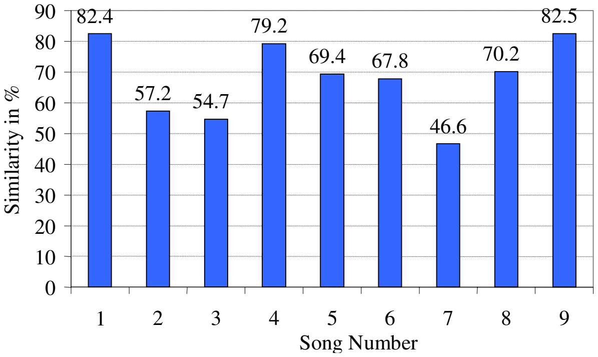


Figure 9: Mean similarities for each tapped query with the referring rhythm.

During the experiments it could be seen that users tend to drum along and try to enhance the rhythm with additional strokes, being only poorly similar with the original. This for example causes the poor similarity of the tapped query with its original for the songs two, three, and seven (Figure 9). The melody of these songs contain long sustained notes which all users reproduced 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.

During the experiments two sorts of errors were predominantly. The users sometimes lost their measure and restarted the tapping of the rhythm at one of the next bars. The test persons often started tapping one bar too soon or too late.

7. FUTURE WORK

The usability of the BeatBank system needs some further investigations, using a much larger database and more search queries. The future experiments will also have to show whether the melodies in a larger database are too much alike, causing too many best match candidates during the search process. Therefore experiments similar to those in [7] need to be carried out.

Further, efficient similarity measures need to be developed, compensating the two frequently occurring typical errors described (loss of measure and early/late start).

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 allows other research groups to test their own similarity measures.

8. REFERENCES

[1] Baeza-Yates, Ricardo A.; Perleberg, Chris H.: *Fast and practical approximate string matching*. In: *Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching*, 1992

[2] Batke, Jan-Mark; Eisenberg, Gunnar; Weishaupt, Philipp; Sikora, Thomas: *A Query by Humming system using MPEG-7 Descriptors*. In: *Proceedings of the 116th AES Conventition*, 2004

[3] Dreyfus, Stuart E.; Law, Averill M.: *The art and theory of dynamic programming*. New York, USA : Academic Press, 1977

[4] Ghias, Asif; Logan, Jonathan; Chamberlin, David: *Query by humming - musical information retrieval in an audio database*. In: *ACM Mulitmedia 95 - Electronic Proceedings*, 1995

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

[6] Jang, Jyh-Shing Roger; Lee, Hong-Ru; Chen, Jiang-Chun: *Super MBox: An Efficient/Effective Content-based Music Retrieval System.* In: *The 9th ACM Multimedia Conference*, 2001

[7] Jang, Jyh-Shing Roger; Lee, Hong-Ru; Yeh, Chia-Hui: *Query by Tapping: A New Paradigm for Content-Based Music Retrieval from Acoustic Input*. In: *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, 2001

[8] Jang, Jyh-Shing Roger; Lee, Hong-Ru; Yeh, Chia-Hui: *A Query-by-Tapping System for Music Retrieval*. In: *Proceedings of the Second IEEE Pacific-Rim Conference on Multimedia*, 2001

[9] Kim, Youngmoo E.; Chai, Wei; Garcia, Ricardo: *Analysis of a contour-based representation for melody*. In: *Proc. International Symposium on Music Information Retrieval*, 2000

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

[11] McNab, Rodger J.; Smith, Lloyd A.; Witten, Ian H.; Henderson, Clare L.; Cunningham, Sally Jo: *Towards the Digital Music Library: Tune Retrieval from Acoustic Input*. In: *Proc. ACM Digital Libraries*, 1996

[12] Paulus, Jouni; Klapuri, Annsi: *Measuring the Similarity of Rhythmic Patterns*. In: *ISMIR 2002 3rd International Conference on Music Information Retrieval*, 2002

[13] Sankoff, David; Mongeau., Marcel: *Comparison of musical sequences*. In: *Computers and the Humanities 24*, 1990

[14] Uitdenbogerd, Alexandra; Zobel, Justin: *Matching techniques for large music databases*. In: *Proceedings of the Seventh ACM International Multimedia Conference*, 1999

[15] Uitdenbogerd, Alexandra L.; Chattaraj, Abhijit; Zobel, Justin: *Music IR: Past, Present and Future*. In: *Proceedings of the International Symposium on Music Information Retrieval*, 2000

[16] Uitdenbogerd, Alexandra L.; Zobel, Justin: *Music ranking techniques evaluated*. In: *Twenty-Fifth Australasian Computer Science Conference*, 2002