Music identification with MPEG-7

Holger Crysandt

Institute of Communications Engineering (IENT)
Aachen University (RWTH)
52056 Aachen, Germany
crysandt@ient.rwth-aachen.de

ABSTRACT

Realtime music identification became more and more interesting within the past few years.[1,2] Possible fields are for example monitoring a radio station in order to create a playlist or scanning network traffic in search of copyright protected material.

This paper presents a client-server application to identify an unknown segment of music. The extraction and exchange of descriptive data is done with MPEG-7 only. This paper also explains how to define the similarity between two segments of music and determents its robustness towards perceptional audio coding and filtering. It also introduces an indexing system to reduce the number of segments which have to be compared to the query.

Keywords: MPEG-7 Audio, music identification, nearest neighbor search, search tree

1. INTRODUCTION

MPEG-1, MPEG-2 and MPEG-4 are well known standards for encoding audiovisual data. The goals of MPEG-7 are completely different.[3–7] It is not another compression standard but a mechanism for extracting and exchanging significant information of multimedia contents.

Significant information of an audio content which can be described with MPEG-7 are[3]:

storage media, location, ...

format file-format, bit rate, size of sequence [bytes]

author/ producer person/ group/ company, title, year, ...

content description loudness, spectral characteristics, harmonicity, ...

segmentation number and length of the tracks on a CD, silence, ...

The *AudioSignature Description Schema* is specialized to describe the content of a piece of music in order to identify it. In spite of its compact size it was shown[8,9] that it is very robust against most common modifications of music signals like:

- filtering (lowpass, highpass, bandpass, equalizer)
- perceptual audio coding (mp3, ogg)
- additive noise
- amplitude change
- compressor, limiter (radio stations)
- random start

Storage and Retrieval Methods and Applications for Multimedia 2004, edited by Minerva M. Yeung, Rainer W. Lienhart, Chung-Sheng Li, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 5307 © 2004 SPIE and IS&T · 0277-786X/04/$15

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

This paper determines the robustness of this *Description Schema* towards perceptual audio coding, equalizer and random start. It introduces the *mahalanobis distance* to specify the similarity between known segments in the database and a segment which has to be identified. Based on this distance an indexing algorithm is introduced which increases the speed of finding the segment which is most similar to the query.

The algorithm mentioned above is implemented on a client-server application to evaluate the performance of the *AudioSignature Description Schema* and the mahalanobis distance towards modifications mentioned above.

2. ROBUSTNESS OF THE DESCRIPTION SCHEMA

The *AudioSignature Description Schema* consists of two matrices called ”Mean” and ”Variance”. Each matrix consists of 16 columns (default) representing 16 frequency sub-bands and approximately 1 row per second (default). The temporal resolution and the number of sub-bands can be varied to increase or decrease the resolution in order to find a good tradeoff between size of the description schema and its robustness.

To create the *AudioSignature DS* the average flatness of the spectrum is determined for each sub-band. Small values of flatness indicate peaks in the spectrum which is typical for tonal components. Values close to one refer to a flat spectrum corresponding to noise-like or impulse-like signal. For the complete extraction process of the *AudioSignature Description Schema* refer to.[10]

2.1. Feature vector

Based on both matrices a feature vector for a segment of music can be defined. For each segment of music a feature vector is created by arranging the values of both matrices by reading the values of the corresponding rows of the first matrix and appending the rows with the same indeces of the second matrix. A segments of 8 seconds for e. g. leads to a feature vector with 256 dimensions.

The order of the values of the feature vector has no influence on the performance of the identification algorithm. Therefor it can be adapted to the memory model of the application. If the matrices are stored colum-major it might be more performant to read the values of the corresponding segment columnwise.

2.2. Similarity between two segments of music

With the feature vectors described above the similarity between two segments can be defined using the mahalanobis distance (1).

$$(q - s_{n,\Delta t})^T R_{xx}^{-1} (q - s_{n,\Delta t}) \tag{1}$$

with:

q: feature vector of query
$s_{n,\Delta t}$: feature vector of songs number n after Δt seconds
R_{xx}: Covariance matrix

$$R_{xx} = \begin{pmatrix} Var(v_1) & Cov(v_1,v_2) & \dots & Cov(v_1,v_n) \\ Cov(v_2,v_1) & Var(v_2) & \dots & Cov(v_2,v_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(v_N,v_1) & Cov(v_N,v_2) & \cdots & Var(v_N) \end{pmatrix}$$

The Covariance matrix R_{xx} is a NxN-matrix (N: length of feature vector). It describes the strength and the covariance of the distortion between each pair of dimensions of the feature vector. It depends on the channel with which the music signal is modified and has to be measured or estimated (Fig. 1).

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

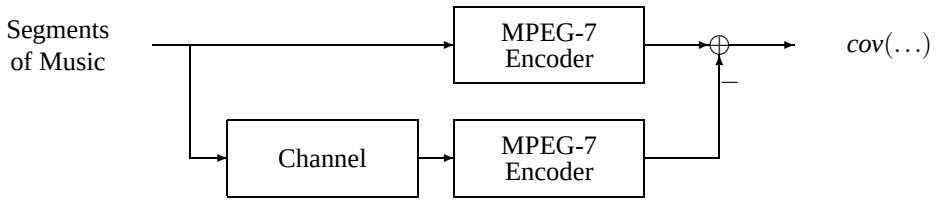

Figure 1. Estimation of the covariance matrix R_{xx}

Frequency sub-bands with a high distortion are weighted low. Therewith the decision is mostly based on the sub-bands with a low distortion. The mahalanobis distance also takes into account the covariance of the errors between different dimensions especially the covariance between successive values of the same sub-band and the correlation between values of adjoining sub-bands.

To find the segment of music in your database which fits best to the query determine the mahalanobis distance between corresponding feature vector of the query and every segment in the database. The segment with lowest distance is probably the segment one is looking for (*nearest neighbor*). A more intelligent to find the best segments will be introduced in the next section.

3. INDEXING SYSTEM

The number of dimensions of the feature vectors (3,4) is currently too high to enable an effective indexing system (*Curse of Dimensionality*). Because of this the size of the feature vector has to be reduced.

The *cholesky decomposition* creates an upper triangular matrix C.

$$R_{xx}^{-1} = C^T C \quad \text{(Cholesky decomposition)} \tag{2}$$

With (2) new feature vectors q' and $s'_{n,\Delta t}$ can be defined:

$$q' = C * q \tag{3}$$
$$s'_{n,\Delta t} = C * s_{n,\Delta t} \tag{4}$$

With those modified feature vectors (3, 4) the *mahalanobis distance* (1) can be simplified to an *Euclidian distance* (5):

$$(q - s_{n,\Delta t})^T R_{xx}^{-1} (q - s_{n,\Delta t}) = \left(q' - s'_{n,\Delta t}\right)^T \left(q' - s'_{n,\Delta t}\right) \tag{5}$$

The Euclidian distance is much less complex than the mahalanobis distance but leads to the same result. Furthermore the Euclidian distance enables a *nearest neighbor* search with ss-trees and kd-trees. These search algorithm are known for years and implemented on several platforms and database systems.

3.1. Compression of Feature Vectors

To get a compressed representation of the modified feature vectors (3, 4) the coordinate system is changed. First the average mean value of each dimension is determined and subtracted from each each value of the dimension. Thereby the center of the coordinate system is moved to the center of gravity of all feature vectors.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

Secondly the axis of the coordinate system are rotated. The transformation matrix is given by using a svd* of all or some randomly selected modified feature vectors. The svd determines the eigenvalues of the feature vectors and calculates a transformation matrix V with which all features are multiplied. This transformation is an orthonormal transformation which means that the *Euclidian distance* of two feature vectors before and after the transformation of the coordinate system remains the same.

As a result the first axis of the coordinate system is set parallel to the direction with the highest variance, the second axes is set parallel to the direction with the second highest variance, ... (Fig. 2).

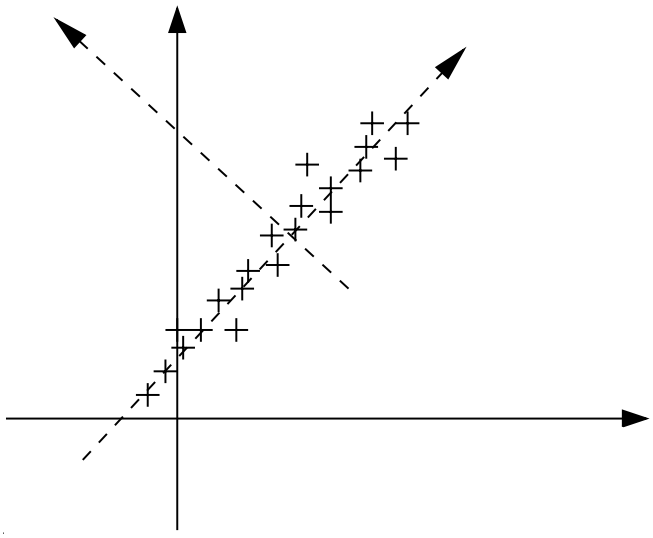

Figure 2. Moving the center of the coordinate system and turning the axis has no effect on the Euclidian distance of two feature vectors

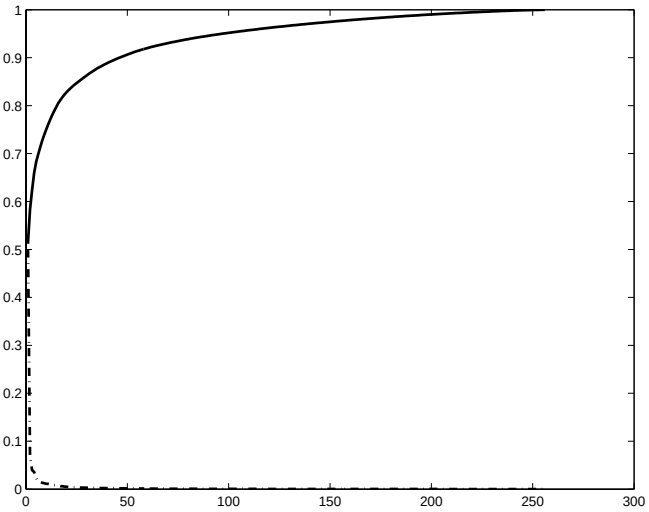

Figure 3. Concentration of signal energy on only a few dimensions of the feature vector

After the transformation the energy of the vectors is concentrated in the first dimensions, whereas the energy of the last

*__svd__: **s**ingular **v**alue **d**ecomposition

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

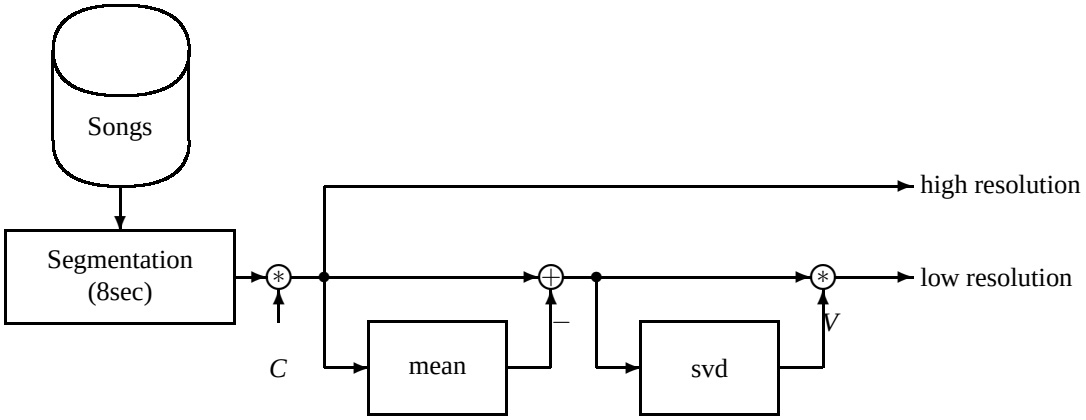

Figure 4. Creation of high and low resolution keys

dimensions is close to zero. Under real conditions more than 50% of the signal energy of the modified feature vectors was concentrated in the first coefficient, 66% of the energy in the first four coefficients and 80% of the energy was concentrated within the first 16 coefficients (Fig. 3).

To get a compressed description of the feature vector, only the first dimensions are stored. The other dimensions are ignored for the low resolution representation. By doing so the low resolution feature vectors can be reduced to 6-12 dimensions. This compression is not lossless but enables an effective clustering of similar feature vectors which is necessary for an efficient *nearest neighbor* search with kd- or ss-trees[11] (section 3.2).

Due to the loss of information performance of the identification decreases. But the low resolution can be used to cluster similar segments in order decrease the number of segments which have to be compared to the query.

3.2. Search Trees

For an efficient *nearest neighbor* search feature vectors close to each other are placed within an object. Objects close to each other are placed in objects again. Commonly used objects are n-dimensional cubes (kd-trees, Fig. 5) or n-dimensional spheres (ss-trees, Fig. 6).

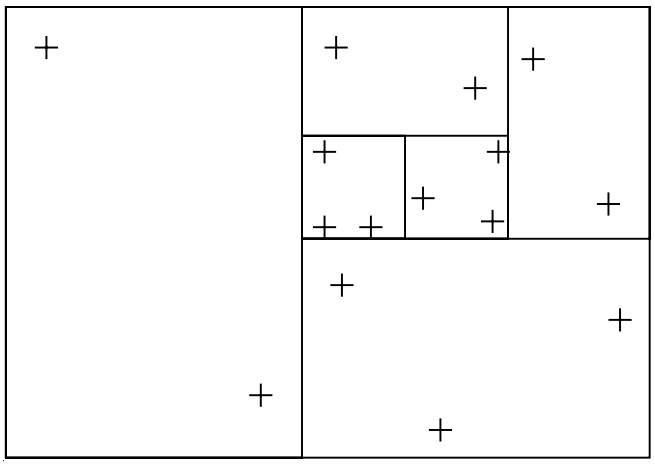

Figure 5. KD-Tree; vectors close to each other placed in n-dimensional cubes

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

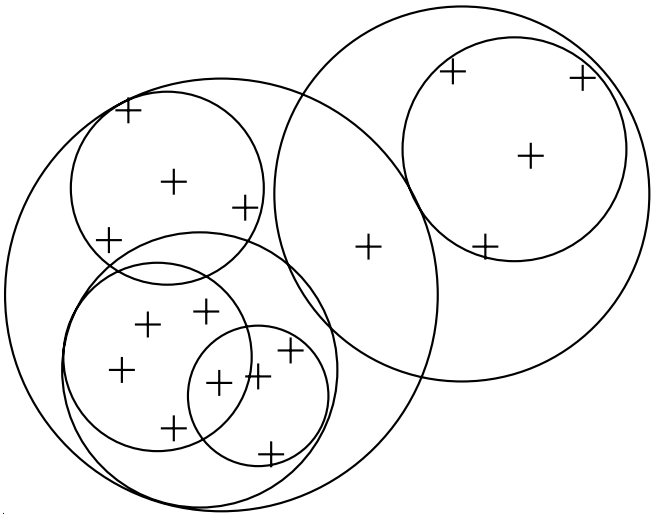

Figure 6. SS-Tree; vectors close to each other placed in n-dimensional spheres

With a low and high resolution representation of the query the search for the segment which fits best is done in two steps: First the covering objects stored in the database are compared to the query using the low resolution representation. If the object is too far away with the low resolution representation from the query the object can be ignored including all its objects inside. But if the object is close to the query or the query is inside the object then the object has to be opened. Inside are either other objects or feature vectors.

If there are object inside the algorithm has to continue recursively and has to calculate the distance to the new objects. Otherwise the feature vectors inside the object are compared to the query using the best resolution available. The search can be terminated when the distance between query and every object left is greater than the distance between query and the closest feature vector.

Note: The minimum distance between query and object is always less or equal to the distance of the query and the feature vector inside of the object. This guarantees that no ”possible candidate” is dropped during the first step of the query.

4. CLIENT-SERVER ARCHITECTURE

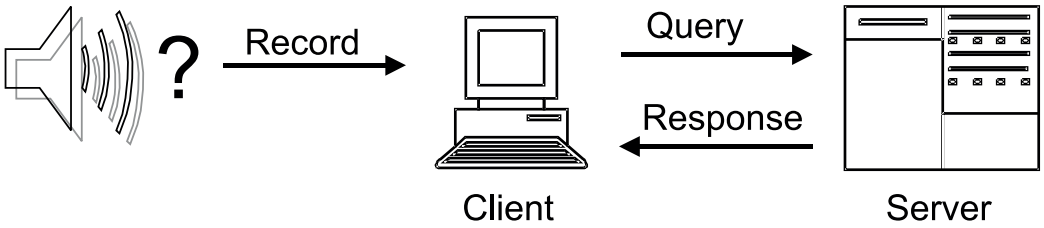

Figure 7. Client-server architecture of the music identification system

To determine the performance of the ideas described earlier the algorithms were implemented in a client-server application (Fig. 7). The server reads a set of MPEG-7 descriptions including name of the artist or band, title of the song and the *AudioSignature DS* of the audio signal. After creating the index tree for all segments of music the server waits for requests

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

from clients.

The client reads the audio signal for about 10 seconds and only creates the *AudioSignature DS* only to keep the MPEG-7 description of the segment as small as necessary. Then it sends the description (not the audio signal) to the server. After a while it receives the response from the server which is in MPEG-7 format, too. It contains the name of the artist or band and the title but it can also include more information like genre or year. It might also be possible to include a link to the original audio file.

5. EXPERIMENTAL RESULTS

The client-server application described above was implemented in Java. For the test a database with 10,000 songs was created[12] which leads to 2,500,000 segments with a length of 8 seconds. The values of the *AudioSignature DS* were stored as 32-bit floating point values. The covariance matrix R_{xx} was measured for the case of perceptual audio coding with random start and random equalizer.

	6	8	12	16
kd-tree	7.86%	8.7%	10.8%	11.7
ss-tree	**6.5**%	6.8%	9.6%	12.4%

Figure 8. Average number of segments [%] which have to be compared to the query for different length of compressed feature vectors and for different search trees

As test set 300 Songs of the database were used. From each song a segment of 8 sec was taken at a random position. Each segment was modified by perceptual audio coding (mp3, 96kHz) and filtering (equalizer, $\pm$10%, random). Under these test conditions all queries were identified correctly.

During the first step of the query (low resolution) approximately more than 90% (8) of all segments of the database were classified as ”no possible candidate”. The lowest percentage of segments which had to be compared to the query was achieved for a low resolution vector length of 6 and ss-trees. The average response time of the server (Athlon 2.4GHz, 1.5Gb) of this configuration was 0.8 seconds. The performance can probably be improved with more intelligent nearest neighbor search algorithm.

6. FUTURE WORKS & CONCLUSION

MPEG-7 provides is very effective mechanism to gather and exchange descriptive data of a multimedia content. The ”Audio Signature Description Schema” is very suitable to describe the signal in order to identify an unknown segment of music. In combination with the mahalanobis distance to define the similarity between two segments of music a recognition rate of approximately 100% can be expected for the case of perceptual audio coding with filtering and random start.

With a test set of 10.000 songs an average response time less than 1 second was evaluated with the prototype software. The quick response time was enabled by the indexing system which reduces the number of segments which have to be compared the query to approximately 6%. It can be expected that this value will decrease with a larger number of songs in the database.

Two major fields of improvement can be figured out: First the covariance matrix R_{xx} needs to be measured for other modifications than perceptual audio coding with random start. Secondly the indexing mechanism can/must be improved in order to be able to handle huge databases with 100.000 or even millions of songs in acceptable time. It might be useful to map the nearest neighbor search algorithm to (sql-)databases. Mapping the nearest neighbor search algorithm to 16- or 8-bit integer values reduces the amount of memory which leads to an increase of the performance especially when memory has to be swapped on the local hard-disk.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms

REFERENCES

1. Shazam. http://www.shazam.com/.
2. Fraunhofer Institute (IIS), "AudioID: Automatic Identification & Fingerprinting of Audio." http://www.iis.fraunhofer.de/ amm/download/audioid.pdf.
3. ISO/IEC JTC1/SC29/WG11 (MPEG), *Information Technology - Multimedia Content Description Interface*. ISO/IEC, international standard 15938 ed., 2001.
4. B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7*, John Wiley & Sons, LTD, 2002.
5. Signal Processing: Image Communication 16, *MPEG-7: A standardised description of audiovisual content*, 2000.
6. Signal Processing: Image Communication 16, *Low-Level musicl descriptors for MPEG-7*, 2000.
7. A. T. Lindsay and J. Herre, "MPEG-7 and MPEG-7 Audio - An Overview," *Journal of the AES* **49**(7/8), pp. 589–594, 2001.
8. 111th AES Convention, New York, *Advanced Audio Identification using MPEG-7 Content Description*, 2001.
9. 112th AES Convention, Munich, Germany, *MPEG-7 Scalable Robust Audio Fingerprinting*, May 2002.
10. ISO/IEC JTC1/SC29/WG11 (MPEG), *Multimedia Content Description Interface - Part 4: Audio*. ISO/IEC, international standard 15938-4 ed., 2001.
11. D. Fend, W. C. Siu, and H. J. Zhang, *Multimedia Information Retrival and Management*, Springer, 2003.
12. "Java MPEG-7 Audio Encoder." http://sourceforge.net/projects/mpeg7audioenc.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/04/2013 Terms of Use: http://spiedl.org/terms