# Audio Thumbnailing Using MPEG-7 Low Level Audio Descriptors

Jens Wellhausen and Michael Höynck

Institute of Communications Engineering
Aachen University
D-52056 Aachen, Germany

## ABSTRACT

In this paper we present an audio thumbnailing technique based on audio segmentation by similarity search. The segmentation is performed on MPEG-7 low level audio feature descriptors as a growing source of multimedia meta data. Especially for database applications or audio-on-demand services this technique could be very helpful, because there is no need to have access to the probably copyright protected original audio material. The result of the similarity search is a matrix which contains off-diagonal stripes representing similar regions, which are usually the refrains of a song and thus a very suitable segment to be used as audio thumbnail. Using the a priori knowledge that we search off-diagonal stripes which must represent several seconds of audio data and that the adjustment of the stripes must be characteristically, we implemented a filter to enhance the structure of the similarity matrix and to extract a relevant segment as a audio thumbnail.

**Keywords:** Audio Thumbnailing, Audio Browsing, Audio Segmentation, Audio Content Description, MPEG-7

## 1. INTRODUCTION

Feature extraction is necessary to handle the large amount of audio data that is available by modern multimedia technology. The upcoming MPEG-7 standard provides audio feature descriptors, which are useful for many applications. For audio browsing applications, as well as for music analysis and music compression an algorithm for audio thumbnailing on the basis of audio segmentation is needed. Using MPEG-7 meta data for this algorithm seems to be very attractive, because there is no need to transfer huge amounts of copyright protected original audio data in order to analyse the structure of a song.

In this paper we propose also a similarity based algorithm to extract segments which are usable as thumbnail. To be independent from the original audio data source, we use an MPEG-7 low level audio descriptor as a feature vector for similariy calculation. The segment of a song we propose to use as audio thumbnail is the refrain. Due to the fact that the refrain of a song is repeated at several positions in a song, segmentation is done by the search for similarity within a song. The measurement for similarity is the Euclidean distance between all *audio spectrum envelope* descriptors within a song provided by an MPEG-7 description. The resulting similarity matrix is post processed to reduce data and distinguish similar regions of the song. One of these regions is extracted as thumbnail.

Within the similarity matrix refrains are represented as off-diagonal stripes surrounded by interferences based on similarity between smaller muscial blocks. For a reliable detection of sections representing the refrain out of the similarity matrix, we assume that the refrain occurs at least three times in the whole song and is at least five seconds long. In this case, the adjustment of the off-diagonal stripes is characteristic which is used in the extraction step.

Using the audio segment description scheme defined by MPEG-7 the information about the location of the thumbnail within a song may be distributed to other applications. For example, to get a good impression of a

Further author information: (Send correspondence to J. Wellhausen)
Jens Wellhausen: E-mail: wellhausen@ient.rwth-aachen.de, Telephone: +49 241 802 7676
Michael Höynck: E-mail: hoeynck@ient.rwth-aachen.de, Telephone: +49 241 802 7677
http://www.ient.rwth-aachen.de

song a web based browsing tools only needs to transfer the refrain, which is usually only about 8% of the whole audio data of a song.

This paper is organized as follows. After the introduction we give a short overview on related work in the field of audio segmentation in section 2 and on MPEG-7 low level audio descriptors in section 3. In section 4 we describe the steps toward refrain detection, followed by a description of thumbnail extraction in section 5. Some experimental results are shown in section 6. In section 7 we annotate some possible applications for this algorithm. Finally we give our concluding summary in section 8.

## 2. RELATED WORK

A lot of related work is done in the field of audio segmentation. Segmentation methods with the intention to classify audio data segments into predefined classes such as speech, music or environmental sounds are already examined.[1] Audio segmentation by looking for abrupt changes in the trajectory of features is also discussed.[2] This algorithm based on difference analysis is suitable for real time segmentation, but it does not compare and label the resulting segments. Similarity based audio segmentation has been examined up to now using a short time fourier transformation or a chroma-based representation for building feature vectors, which are the basis for similarity calulation[3].[4]

## 3. MPEG-7 LOW LEVEL DESCRIPTORS

The MPEG-7 standard is officially called "Multimedia Content Description Interface". It is a means of providing meta-data for multimedia and not an advancement of the well-known compression standards up to MPEG-4.[5,6] The standard provides a set of descriptors (D), which specify features or attributes of multimedia content. The superior structure is given by description schemes (DS), which describe structure of their components, which may be description schemes or descriptors. To allow the extension of the set of description schemes and descriptors, a description definition language (DDL) is provided by the standard.[7] The interface between description and applications is refined by XML (eXtensible Markup Language).

The XML interface for MPEG-7 descriptors allows a hierarchical description of multimedia content. A good example is a music compact disc. From the outside look you have a case with a booklet and a disc. The booklet could be divided into several pages with information about the songs and the artist. If you look closer to the disc, there are several tracks containing music. The next stage is to examine each track and analyze its structure. At this point a refrain based audio thumbnail extraction algorithm is very useful. Smart MPEG-7 based database applications or music-on-demand services could offer these segments as audio thumbnails, afterwards the user may decide if he accepts more download time or even billing to acquire the whole audio data. For storing the resulting audio thumbnail, the *segment* DS is predestinated.

## 4. REFRAIN DETECTION BY SIMILARITY ANALYSIS

The meta data produced by an MPEG-7 encoder contains several feature vectors which are the result of basic signal processing algorithms. In this application, the *audio spectum envelope* descriptor is used as input data.

### 4.1. The *Audio Spectrum Envelope* descriptor

The *audio spectrum envelope* descriptor contains the logarithmic spectrum of an audio signal. It is member of the basic spectral descriptors class of the MPEG-7 standard containing sub-band descriptions of the short-term audio signal spectrum. The log-frequency scaling of the descriptor serves multiple purposes: it gives a very compact description of the signal's spectral content, and it mirrors the approximately logarithmic response of the human ear.

A basic fact is that this descriptor maintains the power relationship with the original signal preserved in the Fourier transform through Parseval's theorem. Since the envelope is a power spectrum, the sum of all spectral coefficients is equal to the power in the analysis window.

To extract this feature, each analysis frame is multiplied by a Hamming window and transformed into the frequency domain using a fast Fourier transform. The frequency resolution is reduced by resampling the

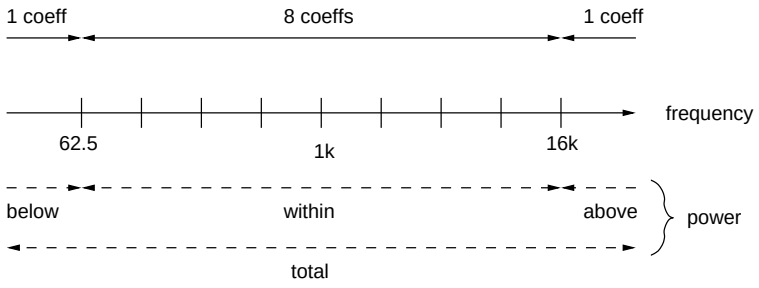


**Figure 1**. Structure of the audio spectrum envelope descriptor

coefficients to a logarithmic scale. Recommendation of MPEG-7 standard is to divide the frequency domain between 62.5 Hz and 16 kHz into eight sub-bands and store one coefficient for each sub-band. Additional two coefficients for the power below 62.5 Hz and above 16 kHz are stored. By default a MPEG-7 audio encoder extracts this feature every 10 ms of audio input data.

The extraction of the *audio spectrum envelope* descriptor results in 10 coefficients every 10 ms for this feature. See Figure 1 for details.

### 4.2. Calculating similarity

The *audio spectrum envelope* descriptor calculated every 10 ms as described above is stored in a time series of feature vectors $V_i$. The number $n$ of vectors can be calculated using the song length $t$ by the ratio $n = \frac{t}{10ms}$.

$$V_i = \begin{pmatrix} v_{i,1} \\ v_{i,2} \\ \vdots \\ v_{i,10} \end{pmatrix} \qquad i = 1, \cdots, n. \tag{1}$$

To detect similar parts of the song, we first determine the similarity between all feature vectors. This results in a similarity matrix $S$ with the dimension $n \times n$.

$$S = \begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nn} \end{pmatrix} \tag{2}$$

Each element $s_{x,y}$ of the symmetric similarity matrix $S$ represents the similarity between the feature vectors $V_x$ and $V_y$. This similarity is defined by the Euclidean norm $d$.

$$d_{x,y} = \sum_{k=1}^{r} (v_{x,k} - v_{y,k})^2 \tag{3}$$

The calculation of distance between feature vectors one by one leads often to an similarity matrix which is unsuitable for refrain detection. Especially in audio sources with significant beat, many discrete feature vectors have a small distance to each other, even when their neighborhoods have bigger distances. To prevent this we finally implemented a diagonal lowpass filter to calculate the distance matrix $S$.

$$s_{x,y} = \sum_{l=1}^{m} d_{x+l,y+l} \tag{4}$$

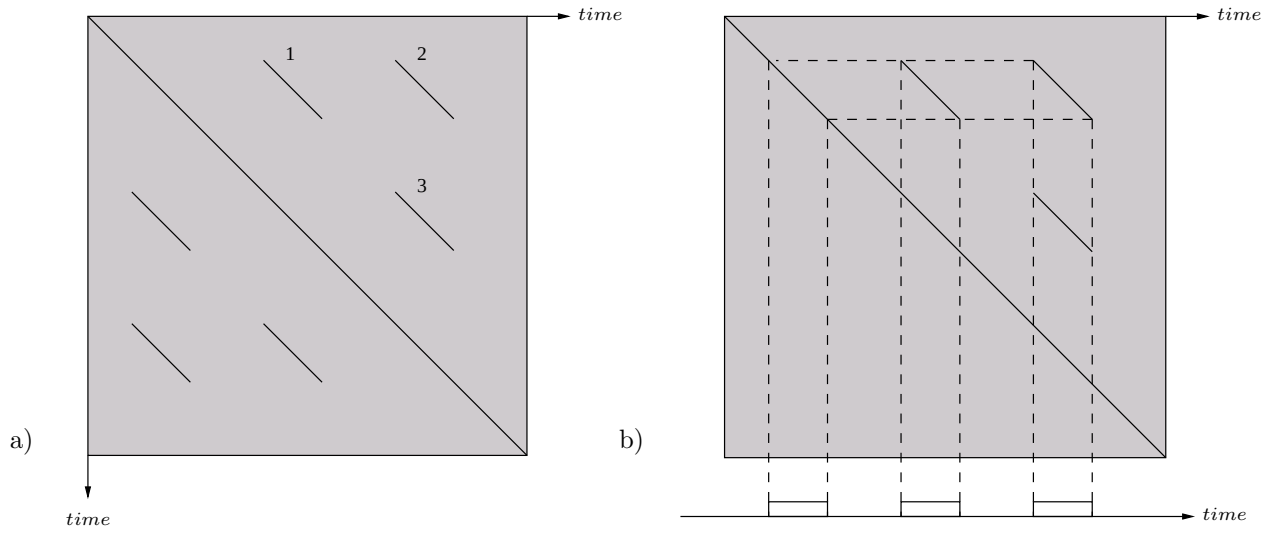


**Figure 2**. **a)** An ideal similarity matrix for a song with three refrains. **b)** Refrain projection to the time line.

In equation (4), the parameter $m$ is the length of the low pass filter. Experiments have shown that for $10 \leq m \leq 15$ the similarity matrix $S$ is most usable for refrain detection.

The parameter $r$ in equation (3) determines how many coefficients of the audio spectrum envelope are used. Experiments have shown that there is not much useful information for calculating similarity in the high-frequency bands above 4 kHz. The distance of a lowpass filtered short cutout of a song to each other block of songs was calculated in a database of approximately 250 songs.

Since the complexity of our algorithm is proportional to $r$, calculation time can be saved. Omitting the frequency-band above 4 kHz (r=7) is a good trade-off between accuracy and complexity of the application. Most relevant information is included in the frequency-bands below 4 kHz.

If the resulting matrix is plotted, similar regions of a song are becoming visible in off-diagonal stripes in a picture disturbed by horizontal and vertical lines. In Figure 2 a) an ideal similarity matrix for a song with three refrains is shown. The main diagonal of this matrix is zero, because each feature vector is equal to itself. The off-diagonal stripe (1) represents the first repetition of the refrain, the stripe (2) represents the second repetition. The first occurrence of the refrain is not represented by an off-diagonal stripe, it is hidden by the main diagonal. The off-diagonal stripe (3) shows the second repetition of the refrain, too, but it arises from the repetition of the second occurence of the refrain. It is not used in the further processing. The projection of the refrains to the time line is shown in Figure 2 b). The begin time and end time of the first occurrence of the refrain is determined by projecting the off-diagonals (1) and (2) to the main diagonal, and then to the time line.

The dimension of the similarity matrix $S$ is $n \times n$, where $n$ is the number of feature vectors. Considering for example a song length of 3 min 20 sec, 750 MB of memory is used when the upper triangular matrix is stored using a 32 bit integer representation. On one hand, it is necessary to calculate this matrix every 10 ms for a reliable refrain detection. Calculating the similarity matrix without using all feature vectors, e.g. using every second or third feature vector, the matrix will not be usable due to interferences based on beat. On the other hand, we do not need a resolution of 10 ms for the extracted thumbnail. If an accuracy of 0.5 s of the detected thumbnail's position is needed, the matrix can be compressed by the factor 50 in both dimensions using a minimum-value filter without loosing essential data.

The similarity matrix S consists of $p^2$ $b \times b$ block matrices.

$$S = \begin{pmatrix} S_{1,1} & \cdots & S_{p,1} \\ \vdots & \ddots & \vdots \\ S_{1,p} & \cdots & S_{p,p} \end{pmatrix} \qquad p = \frac{n}{b} \tag{5}$$

Each block matrix $S_{s,t}$ consists of $b^2$ elements of the matrix $S$.

$$S_{m,n} = \begin{pmatrix} s_{1,1}^{m,n} & \cdots & s_{1,b}^{m,n} \\ \vdots & \ddots & \vdots \\ s_{b,1}^{m,n} & \cdots & s_{b,b}^{m,n} \end{pmatrix} \tag{6}$$

The compressed similarity matrix $\tilde{S}$ is formed using the minimum of each block matrix $S_{m,n}$.

$$\tilde{S} = \begin{pmatrix} \tilde{s}_{1,1} & \cdots & \tilde{s}_{p,1} \\ \vdots & \ddots & \vdots \\ \tilde{s}_{1,p} & \cdots & \tilde{s}_{p,p} \end{pmatrix} = \begin{pmatrix} \min\limits_{k,l}\{s_{k,l}^{1,1}\} & \cdots & \min\limits_{k,l}\{s_{k,l}^{1,p}\} \\ \vdots & \ddots & \vdots \\ \min\limits_{k,l}\{s_{k,l}^{p,1}\} & \cdots & \min\limits_{k,l}\{s_{k,l}^{p,p}\} \end{pmatrix} \tag{7}$$

The diagonal stripes indicating positions of the refrains are detected by image processing algorithms. An automatic edge detection to find the diagonals and extract a thumbnail is performed, but first some postprocessing of the picture has to be done to sharpen the relevant diagonals.

### 4.3. Postprocessing

It is not possible to assume that the diagonals representing the refrain are characterized by absolute minimums in the distance matrix. The reason is that determining the similarity based on the Euclidean distance results in small values when the signal has small power, even when there is no similarity.

The compressed distance matrix is filtered using a priori knowledge. The refrains are represented by off-diagonal stripes which must have a length containing severals seconds of audio data, and the surrounding area of these stripes are characterized by high signal power. Taking a block of the matrix with the dimension $d \times d$, the mean value of the main diagonal is divided by the mean value of the whole block. The results of this operation are stored in the similarity matrix for all coordinates.

$$m_{diag,x,y} = \frac{\sum\limits_{i=0}^{d} \tilde{s}_{x+i,y+i}}{d} \qquad m_{block,x,y} = \frac{\sum\limits_{i=0}^{d}\sum\limits_{j=0}^{d} \tilde{s}_{x+i,y+j}}{d^2} \tag{8}$$

$$\tilde{s}_{x,y} = \frac{m_{diag,x,y}}{m_{block,x,y}} \tag{9}$$

This operation performs a sufficient enhancement of diagonal stripes with the minimum length $d\sqrt{2}$. Horizontal and vertical edges are eliminated. It is easy then to extract the off-diagonal stripes and determine the beginnings and endings of the refrains. The first occurrence of the refrain can be determined projecting the off-diagonal stripes onto the main diagonal.

In Figure 4 two examples of similarity matrices after the postprocessing step are shown. In Figure 4(a) it is from Boney M's "Rasputin", in Figure 4(b) it is from Metallica's "Prince Charming". The off-diagonal stripes representing refrains are marked with boxes.

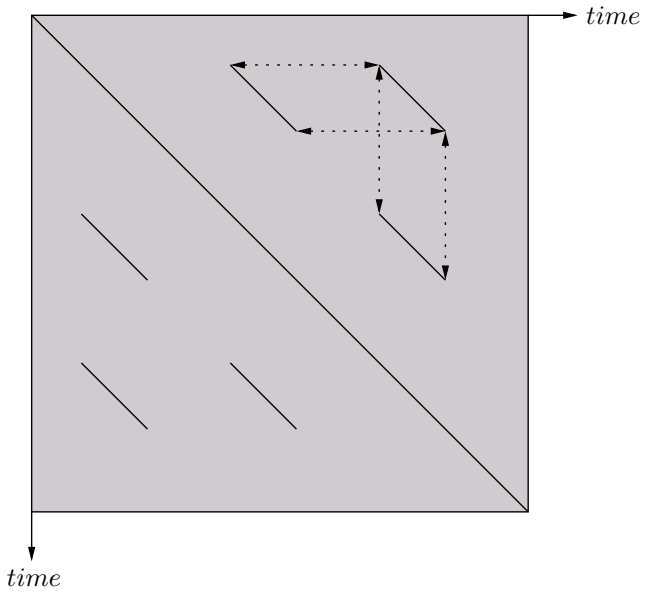


**Figure 3**. Extraction of refrain segments. Required alignment of off-diagonal stripes for a song with three refrains.

## 5. THUMBNAIL EXTRACTION

Finding a refrain within the similarity matrix to be used as thumbnail is the last step of the process. Up to now, the similarity matrix contains off-diagonal stripes representing repetitions of similar phrases and thus refrains. The task of this step is to identify these stripes. Additionally, in most cases there are here and there smaller off-diagonals due to self-similarity of smaller sections. Our algorithm must be smart enough to decide whether a off-diagonal stripe represents a refrain or just a smaller block which is similar to any other block.

We have defined two properties which are the basis for this decision. The first one is that we only search for off-diagonal stripes which are at least five seconds long. This very easy property suppresses small off-diagonal stripes from detection, which often occur when there are repeated significant instrument riffs or beat events.

The second property for the decision whether a off-diagonal stripe is a refrain or not is the adjustment within the matrix. Assuming that the refrain occurs at least three times, the off-diagonal stripes must be arranged in a triangular shape as shown in Figure 3. As mentioned in section 4, these off-diagonal stripes represent (from left to right) the first repetition of the first refrain and the second repetition of the first refrain. The off-diagonal stripe underneath represents the repetition of the second refrain, thus the third refrain.

If a structure as mentioned above is found within the similarity matrix, our algorithm declares one of the found segments as audio thumbnail. An appearance of the refrain of more than three times is compatible to this decision.

## 6. EXPERIMENTAL RESULTS

The first step after which a visualization of results is reasonable is when the post processed similarity matrix is available. We picked deliberately two examples of very different genre out of a huge database, Boney M's "Rasputin" and Metallica's "Prince Charming". The appropriate matrixes are visualized in Figure 4.

In the first example (Figure 4 **a)**), the extraction of the audio thumbnail is very easy due to the clear structure of the off-diagonal stripes. Our extraction algorithm delivers us a segment of 27 seconds length starting at 80 seconds playtime. With an accuracy of one second this result is identical to a hand-crafted refrain extraction.

Within the second example (Figure 4 **b)**) it is more complicated to extract a off-diagonal stripe which is useable as a thumbnail. If we have a closer look to the off-diagonal stripe on the left side we see a continuation of

a) Metallica - Prince Charming

| refrain no. | automatic extraction | | manual extraction | |
|---|---|---|---|---|
| | start time | end time | start time | end time |
| 1 | 80 s | 107 s | 86 s | 108 s |
| 2 | 137 s | 165 s | 148 s | 170 s |
| 3 | 269 s | 299 s | 279 s | 301 s |

b) Boney M - Rasputin

| refrain no. | automatic extraction | | manual extraction | |
|---|---|---|---|---|
| | start time | end time | start time | end time |
| 1 | 81 s | 96 s | 80 s | 96 s |
| 2 | 144 s | 159 s | 143 s | 159 s |
| 3 | 230 s | 245 s | 228 s | 244 s |
| 4 | 245 s | 255 s | 244 s | 260 s |

**Table 1.** Automatically extracted refrains vs. hand-crafted results
**a)** Metallica - Prince Charming **b)** Boney M - Rasputin

the stripe to the upper left. This is because of the special structure of the song. Before the first and the second occurence of the refrain, a segment with the same melody but different text is located, which is represented by this extension of the off-diagonal stripe. At this point our algorithm described in section 5 detects that this segment does not belong to the refrain, because it can not be found in the beginning of the third repetition of the refrain. The extraction algorithm delivers a segment of 15 seconds length starting at 81 seconds playtime.

In Table 1 the automatically extracted refrains are faced with hand-crafted results.

## 7. APPLICATIONS

The most important application of audio thumbnailing are multimedia databases, where convenient browsing tools are needed. Especially the fact that our algorithm is based on MPEG-7 meta data makes it applicable for audio-on-demand services. For example, the user can download the obviously copyright free MPEG-7 meta data, his local tools can analyse the material and request a small segment as thumbnail. Only if the result appeals to the user, he may download the entire audio data file.

Another interesting application is considered to be possible if original audio data and concerning meta is available in parallel. Music players now could be equipped with a music scan function. The user would be able to get a quick overview about the content of his music medium. Todays compact disc players often have an intro-scan function which plays the first few seconds of a song with this purpose, but the refrain as thumbnail would be a better choice for this task.

## 8. CONCLUSION

A new audio thumbnailing algorithm based on the MPEG-7 *audio spectrum envelope* descriptor was introduced. First, a similarity matrix is calculated where similar regions of a song become visible in off-diagonal stripes. After postprocessing to improve sharpness of the off-diagonal stripes and in consideration of the common structure of songs, similar regions that represent refrains are extracted. The time information for the extracted regions are used to mark a section of a song as thumbnail.

Depending on the user interface and system, there are several ways of making audio thumbnailing available. First, software audio players could analyse Another idea especially for web based applications or for portable devices is to store time information where a thumbnail within an audio file can be found as meta data into audio files. For example, the ID3 tag of MP3 files provides timing codes,[8] which seem to be predestinated to store such kind of information.

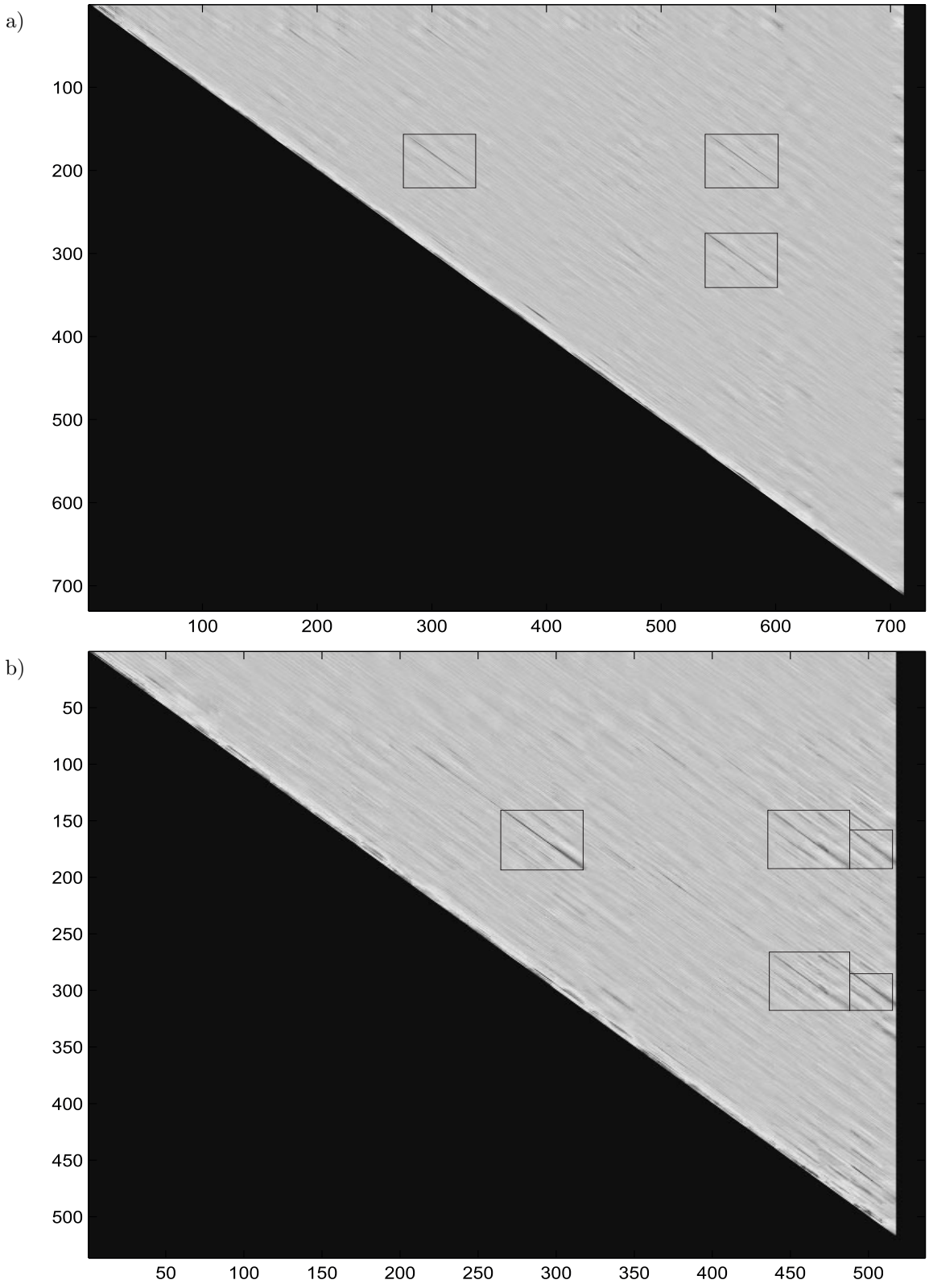


**Figure 4**. Similarity matrixes **b)** Metallica - Prince Charming and **a)** Boney M - Rasputin

## REFERENCES


1. H. Jiang, T. Lin, and H. Zhang, "Video segmentation with the support of audio segmentation and classification," tech. rep., Microsoft Research, China.
2. H. Sundaram and S.-F. Chang, "Video scene segmentation using video and audio features," Dept. of Electrical Engineering, Columbia University, New York.
3. J. T. Foote and M. L. Cooper, "Media segmentation using self-similarity decomposition," in *Storage and Retrieval for Media Databases*, SPIE, 2003.
4. M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, New York), Oct 2001.
5. R. Koenen and F. Pereira, "MPEG-7: A standardised description of audiovisual content," in *Signal Processing: Image Communication*, **Vol. 16**, pp. 5–13, Elsevier, 2000.
6. A. T. Lindsay and J. Herre, "MPEG-7 and MPEG-7 audio - an overview," *J. Audio Eng. Soc.* **Vol. 49**, pp. 589–594, July/August 2001.
7. B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7*, John Wiley & Sons, LTD, 2002.
8. "http://www.id3.org."