Histogram matching for music repetition detection

Conference Paper · June 2009

DOI: 10.1109/ICME.2009.5202583 · Source: DBLP

Dong Wang
East China University of Science and Technology
72 PUBLICATIONS **774** CITATIONS

Tong Zhang
McMaster University
80 PUBLICATIONS **469** CITATIONS

HISTOGRAM MATCHING FOR MUSIC REPETITION DETECTION

Aibo Tian[1*], *Wen Li*[1*], *Linxing Xiao*[1], *Dong Wang*[2], *Jie Zhou*[1] *and Tong Zhang*[3]

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
[1]Department of Automation, [2]Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
[3]Hewlett-Packard Laboratories
1501 Page Mill Road, Palo Alto, CA 94304, USA

ABSTRACT

Repetition detection is a fundamental issue for music thumbnailing and summarization. In this paper, we propose a new feature, called chroma histogram, which enables us to find out repetitive segments from popular songs accurately and quickly. The feature is robust to tempo variation, because sequential information is removed during the process. The low dimensional feature guarantees a very low computational cost, which is proved by theoretic analysis and experimental evaluation. The objective evaluation results demonstrate that our algorithm outperforms previous approaches in terms of both detecting accuracy and efficiency.

Index Terms— Repetition detection, Music structure analysis, Histogram, Pattern matching

1. INTRODUCTION

With the increasing amount of available digital music, how to effectively and efficiently browse music has become a challenge. To address this challenge, many researches have been exploited to extract the most representative parts of a song, which can be used as music thumbnails and summaries. As one fundamental issue of such applications, repetition detection attracts the interests of many researchers.

Considerable approaches are reported to use different features for repetition detection. Cooper and Foote [1] proposed to use the Mel-frequency Cepstrum Coefficients (MFCCs) to extract the segments in a song which are similar in timbre. Jensen, Xu and Zachariasen proposed to use rhythm-related feature in [2]. Recently, many researches [3, 4, 5] focused on the pitch-related features, such as chroma, to detect melody based repetitions which fit the human perception of music similarity. Given a chroma vector sequence, Zhang and Samadani [5] divided it into several fixed-length segments with a fixed hop size between two consecutive ones. To ensure the efficiency of their system, the authors directly used Euclidean distance to measure the similarity of two segments. However, the Euclidean distance based matching has a very strict sequential constraint, which requires frame-to-frame matching of two segments. In real applications, mismatching problems may be caused by such sequential constraint in two cases. First, the division cannot ensure that the boundaries of repetitive segments are matched exactly, because of different time intervals between repetitive pairs. Second, contents in the repetitions are not matched frame by frame, because of tempo variation. In [4], dynamic programming was introduced to solve the mismatching problems. The dynamic programming based matching can loosen the strict sequential constraint, while remaining order information, by finding an optimal nonlinear alignment between two segments. However, the heavy computational cost makes this algorithm not applicable for on-line repetition detection. The problems in [5] and [4] lie in how to deal with the sequential information. Neither of them well balance the effectiveness and efficiency. So we propose to find a new method, removing the sequential constraint.

Our method is based on the bag of words model. This model is a well known method which only focuses on the words distribution, disregarding the orders. It is often used in natural language processing [6]. Recently, Csurka [7] and Fei-Fei [8] proved that this model had good performance on visual categorization and scene classification in computer vision.

In this paper, we propose a novel histogram matching method based on Chroma feature to address the problems mentioned above. The flowchart of the proposed method is shown in Fig. 1. Different from previous works which directly use the chroma vector sequence as representation of each segment, our method extracts a statistic feature called chroma histogram to represent each segment. Histogram is a very powerful representation which maps a set of features to a vector of bins with fixed length by counting number of features falling into each bin. As a holistic and unstructured representation, the histogram removes sequential information of frames in one segment, thus is able to handle the mismatching problem caused by strict sequential constraint. Furthermore, the efficiency of the distance calculation between two histograms guarantees the very low computation cost of our method. The experimental results show significant improvements to previous works both on effectiveness and efficiency.

The remainder of this paper is organized as follows. In Section 2 we describe the details of the histogram feature. Section 3 summarizes the procedure of repetition detection based on histogram matching. Section 4 presents the complexity analysis of the proposed method. Experimental results are shown in Section 5, and Section 6 concludes the paper.

2. FEATURE EXTRACTION

By converting a chroma vector sequence into a fixed-length vector, histogram removes the sequential information while keeps the chroma composition information. In our work, a histogram bin corresponds to a particular region in the chroma feature space, which is pre-determined by clustering. And the value of each bin is the num-

[*] These authors contributed equally to this work

978-1-4244-4291-1/09/$25.00 ©2009 IEEE

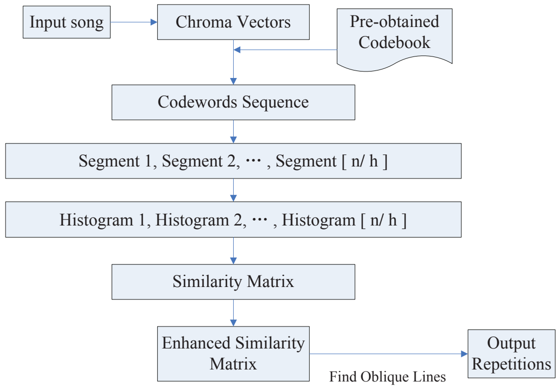

Input song
Chroma Vectors
Pre-obtained Codebook
Codewords Sequence
Segment 1, Segment 2, ⋯ , Segment [n/ h]
Histogram 1, Histogram 2, ⋯ , Histogram [n/ h]
Similarity Matrix
Enhanced Similarity Matrix
Find Oblique Lines
Output Repetitions

Fig. 1. The flowchart of the proposed method. n and h represent the length of chroma vector sequence of the whole song and the hop.

ber of chroma features falling into the particular region. Details of feature extraction are described in the following.

2.1. Basic Feature Extraction

To find out repetitions in melody, we use chroma feature, which reflects power distribution of the 12 pitch classes, as the basic feature, according to the algorithm proposed in [9] . One difference between [9] and other chroma calculating algorithms is that it uses instantaneous frequency within each Fast Fourier Transformation (FFT) bin to locate strong tonal components in the spectrum and to get a higher resolution estimation of the frequency. Experiments show better performance of [9] than other algorithms. Given a song, we divide it into a series of 92ms frames with 3/4 overlap and extract the 12-element chroma vector for each frame.

2.2. Codebook

Forming codebook is a crucial step in our work, which defines the regions in feature space corresponding to the bins in histogram. Although there are countless chroma vectors whose 12 elements are continuous variables, they can actually be clustered in a finite number of regions in the feature space. This can be proved by music theoretical evidence: the combinations of pitches, like chords but not exactly, in a song cannot be selected freely, because inappropriate combinations will sound like noise. In fact, the number of harmonious combinations is limited. Therefore, we can use clustering methods to find the implicit congregating regions in the feature space. All chroma vectors in the same cluster are regarded as having the same chroma type. And the centers of obtained clusters are recorded as a codebook to represent their corresponding chroma types. In this work, we randomly choose 20 songs, from which nearly 200,000 normalized chroma features are extracted. Then all of the chroma vectors are clustered into 40 clusters using the K-means algorithm. The 40-center codebook is shown in Fig.2. We also try the numbers of clusters from 20 to 220, with the interval of 20. It turns out that 40 is the best choice in Fig. 5.

2.3. Chroma Histogram

As discussed in Section 1, previous methods did not find a good way to deal with the sequential information, so they cannot balance the effectiveness and efficiency. We propose to remove the sequential information by using histogram. In this way, the proposed method is robust to tempo variation. The complexity is also reduced by lower-dimension histogram matching. For a given song, we compute the chroma histogram of each segment instead of the whole song. Following is the detailed procedure in our work:

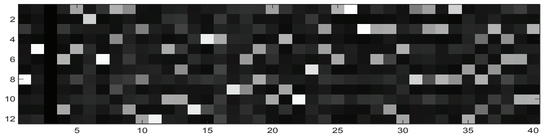


Fig. 2. 40-center codebook. X-axis is their indices representing chroma type. Y-axis represents the 12 pitch classes.

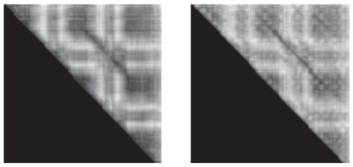


Fig. 3. Comparison of similarity matrix. The left one is obtained without dividing step. The right one is obtained with that step. The ends of the oblique lines are much clearer in the right one.

Step1 The chroma vector sequence is extracted from one song, and each vector is compared with the codebook to get a label. In this case, the 12 × n chroma vector sequence is transformed into 1 × n codewords sequence.

Step2 The whole codewords sequence is divided into a number of 8-second segments with 7-second overlap between consecutive segments.

Step3 For each segment, we divide it into 2 parts with equal length.

Step4 The 40-bin histograms of codewords are obtained for each part then concatenated to form a 1 × 80 vector as the representation of that segment.

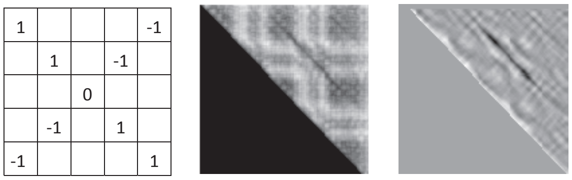
It should be noted that in ***Step3*** and ***Step4*** we do not calculate the histogram for the entire segment directly, but divide it into different parts first. The final obtained vector for a segment is equivalent to a series of ***orderly*** aligned vectors corresponding to the divided parts. Therefore, this representation contains some sequential information, which is proved by Liu [10] to be helpful to improve the performance of histogram algorithm. Fig.3 shows that sequential information can make the repetitions more distinguishable. However, the number of parts is not easy to choose, because little sequential information will improve performance, while excessive one will make the detection sensitive to tempo variation. We evaluate the performances of different *part number* and find that 2 is the best one in Fig. 5.

3. REPETITION DETECTION

Previous methods [3] [5] proposed to detect repetitions by finding out the oblique lines in the similarity matrix. Their results proved that this method for detection is effective, so we use a similar method as theirs while making some improvements. We calculate the similarity matrix based on chroma histogram, and enhance the matrix to make the oblique lines much clearer. Then we propose to find out the repetitions based on the oblique lines.

3.1. Enhanced Similarity Matrix

In this step, the similarity matrix is obtained by calculating the distance between chroma histograms of all pairs of segments using histogram intersection measurement.



1				-1
	1		-1	
		0		
	-1		1	
-1				1

Fig. 4. The left figure is the mask used for enhancement. Grids without figure refer to 0. The middle and right ones are similarity matrices before and after enhancement.

However, the oblique lines, which will be used for repetition detection, in the similarity matrix are not very clear. This is because of the strong variations of songs. Even if two parts are perceived as repetitions, there are still many differences between them. Thus the similarity matrix should be enhanced before being used for repetition detection.

To enhance the oblique lines, we define a 45-degree rotated cross mask to transform all the elements in the similarity matrix, which is shown in Fig. 4. Then for each element in the similarity matrix, the center of the mask is located at it and its value is replaced by the weighted sum of all elements in the mask. This mask can get rid of the rectangular block and enhance the black 45-degree oblique lines. Fig. 4 demonstrates the effect of this mask on similarity matrix.

3.2. Oblique Line Finding

After obtaining the enhanced similarity matrix, Zhang's method [5] is used to find out the repetitions. The four lowest minima of each row and each column in the enhanced similarity matrix, which meet the amplitude and sharpness criteria, are selected, based on the assumption that a paragraph is repeated at most five times in a song.

To reduce the noise, we only record the repetitive pairs longer than 10 seconds. Then a single repetitive pair is treated as a group. And different groups can be merged if they have a common segment. Besides, since our approach aims at detecting repetitions in paragraph level(10-second level), rather than frame level, and a pop song is less possible to have more than five repetitive groups in paragraph level, we only select the five longest groups as the final result.

4. COMPLEXITY ESTIMATION

To prove the efficiency of our method, we estimate the complexity of the proposed method and compare it with [5]. Since the complexity of [4] is much higher than ours, we do not compare with it. In this step, we exclude the complexity of basic feature extraction and codebook formation. Before the estimation, we define some notations.

The length of chroma vector sequence of a segment, the hop between consecutive segments and the whole song are denoted as l, h and n. Let m represents the *cluster number*. And p is defined as the *part number* in Section 2.3 .

4.1. The Proposed Method

The computational cost of each step is listed below.
Transforming the chroma vector sequence to codewords sequence:

$$C_1 = 12 \times m \times n.$$

Computing the histogram for all segments:

$$C_2 = l \times \frac{n}{h}.$$

Calculating the similarity matrix:

$$C_3 = m \times p \times \frac{1}{2} \times (\frac{n}{h})^2.$$

Enhancing the similarity matrix:

$$C_4 = 25 \times \frac{1}{2} \times (\frac{n}{h})^2.$$

Finding out the oblique lines:

$$C_5 = \frac{1}{2} \times (\frac{n}{h})^2.$$

The complexity of the proposed method is the sum of items above:

$$C_p = \sum C_i = (\frac{mp+25}{2h^2})n^2 + (12m + \frac{l}{h})n + O(1).$$

According to the parameters of our method, FFT window of chroma feature is 2048, segment length is 8 seconds, hop size is 1 second, part number is 2 and cluster number is 40. Assuming a song is 4-minute long with sample rate 22050Hz, we can approximately set the parameters as l=320, h=40, n=10000, m=40, p=2, and obtain $C_p \approx 8.2 \times 10^6$.

4.2. Zhang's Method [5]

To calculate the similarity matrix, the time cost of Zhang's method [5] is

$$C_1 = 12 \times l \times \frac{1}{2} \times (\frac{n}{h})^2.$$

To find out the oblique lines, the time cost is

$$C_2 = \frac{1}{2} \times (\frac{n}{h})^2.$$

The complexity of [5] is the sum of items above:

$$C_Z = \sum C_i = (\frac{12l+1}{2h^2})n^2 + O(1).$$

According to the parameter setting of [5], FFT window of chroma feature is 4096, segment length is 5 seconds, hop size is 1.25 seconds. Assuming a song is 4-minute long with sample rate 22050Hz, we can approximately set the parameters as l=100, h=25, n=5000, thus $C_Z \approx 2.4 \times 10^7$.

The above analysis shows that the proposed algorithm is much faster than Zhang's method [5], whose efficiency have been proved. The reason is that, the chroma histogram representation significantly reduces the computational cost of the similarity matrix calculation step.

5. EXPERIMENTAL RESULTS

In this experiment, we evaluate the performance of the proposed method and compare it with the algorithms in [4] [5], which are dynamic programming based matching and Euclidean distance based matching respectively. This experiment is carried out using the computer with CPU of AMD Athlon 64 × 2 Dualcore Processor 3600+, and memory of 1GB. The programming environment is matlab 2007.

5.1. Data Set

Our experiment is based on a set of 69 songs, including English pop songs and pure instrumental compositions, from more than 50 artists. All the songs are mono and sampled at 22050Hz. The repetition groups are annotated manually as the ground truth. Each group is at least 10-second long.

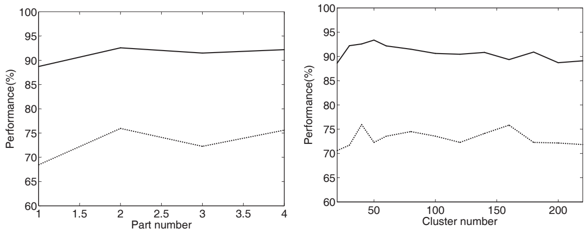


Fig. 5. Experimental results of different parameters. The left one is about part number, and the right one is about cluster number. Solid line is the recall, and dash line is the precision.

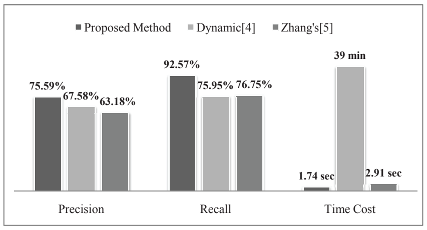


Fig. 6. Comparative results of the proposed method, the dynamic programming method[4] and Zhang's method[5].

5.2. Evaluation Criteria

For a single song, we use the similar definitions of precision rate and recall rate as [11] to measure the quality of our results. They are defined below:

$$\begin{aligned} precision &= \frac{\#\, pos.\, det.\, seg.}{Total\#\, det.\, seg.} \\ recall &= \frac{\#\, det.\, anno.\, seg.}{Total\#\, anno.\, seg.}, \end{aligned}$$

where $\#\, pos.\, det.\, seg.$ is the number of correctly detected segments, $\#\, det.\, anno.\, seg.$ is the number of detected annotated segments, $Total\#\, det.\, seg.$ is the total number of detected repetitive segments, and $Total\#\, anno.\, seg$ is the total number of annotated repetitive segments. They are defined as follows:

Segments are chosen in pair from a detected repetition group, and compared with the annotated ground truth. If they are real repetitions, which means they have an overlap with an annotated ground truth repetitive pair longer than 5 seconds, label them as *positive*, and label the corresponding annotated segments as detected ones. Otherwise, do nothing.

Finally, the precision and recall of the whole data set are calculated as the average of all songs'.

5.3. Result

We first conduct an experiment to tune the parameters (*cluster number* and *part number*). First the *part number* is fixed to 2, and the *cluster number* varies from 20 to 220. As shown in Fig. 5, the best performance is achieved when the *cluster number* equals to 40. And then the *cluster number* is fixed to 40 and the *part number* increases from 1 to 4. As shown in Fig. 5, the performance with *part number* equals to 2 outperforms others.

With the best parameter setting, we evaluate the performances in three aspects: precision, recall and speed. Comparative results of our method and those in [4, 5] are shown in Fig. 6.

It shows that the proposed approach achieves 75.59% of precision rate and 92.57% of recall rate, which exceed the second best one 8.01% and 15.82%, respectively. Considering the complexity of the dynamic method, We allows it to search for only three points in each step, and the time cost is already about 39min per song. This method is supposed to be more accurate if more points are allowed in the searching, however, the time cost would raise as well. Comparatively, our algorithm uses 1.74s on average processing a song, which is nearly 60% of Zhang's 2.91s per song. The percentile is smaller than the complexity calculation in Section 4, because the time cost is not only determined by complexity under the matlab environment. These experimental results prove the good performance of the proposed method both in quality and speed.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new feature, called chroma histogram, for repetition detection. This feature maps the whole space of chroma vector into a fixed number of regions, which can reflect the types of chroma vector. Since the histogram removes the sequential information of frames in a segment, it is able to handle the mismatching problem caused by strict sequential constraint. Besides, the fixed number of bins guarantees the low complexity of our method.

The experimental results show that our method exceeds the second best one by 8.01% in precision rate and 15.82% in recall rate. Moreover, our method is faster than any other one. These results prove our promises of effectiveness and efficiency, and ensure the feasibility of this approach in on-line systems. For future work, we intent to extend this chroma histogram feature to more applications, such as query-by-humming retrieval of songs.

7. REFERENCES

[1] M. Cooper and J. Foote, "Automatic music summarization via similarity analysis," in *Proceedings of ISMIR*, 2002.

[2] K. Jensen, J. Xu, and M. Zachariasen, "Rhythm-based segmentation of popular chinese music," in *Proceedings of ISMIR*, 2005.

[3] M.A. Bartsch and G.H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proceedings of WASPAA*, 2001.

[4] W. Chai and B. Vercoe, "Music thumbnailing via structural analysis," in *Proceedings of ACM international conference on Multimedia*, 2003.

[5] T. Zhang and R. Samadani, "Automatic generation of music thumbnails," in *Proceedings of ICME*, 2007.

[6] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of ECML*, 1998.

[7] G. Csurka C. R. Dance L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, at ECCV*, 2004.

[8] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of CVPR*, 2005.

[9] D.P.W. Ellis and G.E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programing beat tracking," in *Proceedings of ICASSP*, 2007.

[10] X. Liu D. Wang, J. Li and B. Zhang, "The feature and spatial covariant kernel: Adding implicit spatial constraints to histogram," in *Proceedings of CIVR*, 2007.

[11] M. Levy K. Noland and M. Sandler, "A comparison of timbral and harmonic music segmentation algorithms," in *Proceedings of ICASSP*, 2007.