Large Scale Similar Song Retrieval using Beat-aligned Chroma Patch Codebook with Location Verification

Yijuan Lu[1] and Joseph E. Cabrera[2]
[1]*Texas State University, Department of Computer Science, San Marcos, Texas, U.S.A.*
[2]*Texas A&M University, Department of Computer Science, College Station, Texas, U.S.A.*

Keywords: Music Information Retrieval, Similar Song Search.

Abstract: With the popularity of song search applications on Internet and mobile phone, large scale similar song search has been attracting more and more attention in recent years. Similar songs are created by altering the volume levels, timing, amplification, or layering other songs on top of an original song. Given the large scale of songs uploaded on the Internet, it is demanding but challenging to identify these similar songs in a timely manner. Recently, some state-of-the-art large scale music retrieval approaches represent songs with a bag of audio words by quantizing local features, such as beat-chroma patches, solely in the feature space. However, feature quantization reduces the discriminative power of local features, which causes many false audio words matches. In addition, the location clues among audio words in a song is usually ignored or exploited for full location verification, which is computationally expensive. In this paper, we focus on similar song retrieval, and propose to utilize beat-aligned chroma patches for large scale similar song retrieval and apply location coding scheme to encode the location relationships among beat-aligned chroma patches in a song. Our approach is both efficient and effective to discover true matches of beat chroma patches between songs with low computational cost. Experiments in similar songs search on a large song database reveal the promising results of our approach.

1 INTRODUCTION

Recent years have witnessed the explosive growth of songs available on the Internet. Many sites such as YouTube allow users to upload different kinds of music including songs. Providing accurate and efficient search throughout a large-scale song dataset is a very challenging task. Current song retrieval systems including Google and Youtube commonly utilize the textual information such as the surrounding text for indexing. Since textual information may be inconsistent with the audio content in songs, content-based song search is desired and has been attracting increasing attention.

In content-based song search, a hot topic is similar song retrieval. Similar songs are referred to as the songs, part of which are usually cropped from the same original song, but modified by altering the volume levels, timing, amplification, or layering other songs on top of another, which are commonly known as remixes. Given a query song, the challenge is to find similar versions of the query song in a large web song database. Applications of such a system include finding out where a song is derived from and getting more information about it, tracking the appearance of a song on the Internet, detecting song copyright violation, and discovering modified or edited versions of a song.

Identification of direct similar songs in this content is trivial since the wave properties of the direct similar song can easily be used to locate the original song. However many times users will modify the original song in a way that makes detection of original song from the similar song much more difficult. For example, many times some songs are remixed together, which introduce a large amount of noise to the original song.

In recent years, large scale music retrieval (Bertin-Mahieux et al., 2010; Casey and Slaney, 2007; Maddage et al., 2004; Seyerlehner et al., 2008) with local features has been significantly improved based on Bag-of-Audio-Words (BOAW) model. BOAW model achieves scalability for large-scale music retrieval by quantizing local features (such as beat-chroma patch) to audio words and applying inverted file indexing to index songs via

Lu Y. and E. Cabrera J..
Large Scale Similar Song Retrieval using Beat-aligned Chroma Patch Codebook with Location Verification.
DOI: 10.5220/0004129802080214
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications and Wireless Information Networks and Systems* (SIGMAP-2012), pages 208-214
ISBN: 978-989-8565-25-9

the contained audio words. Although BOAW model makes it possible to represent, index, and retrieve songs like documents, it suffers from audio word ambiguity and feature quantization error. Those unavoidable problems greatly decrease retrieval precision and recall, since different features may be quantized to the same audio word, causing many false local matches between songs. And with the increasing size of song database (e.g. greater than one million songs) to be indexed, the discriminative power of audio words decreases sharply.

To reduce the quantization error, one solution is to utilize location information of local features in songs to improve retrieval precision, since the location relationship among audio words plays a key role in identifying songs. However, how to perform location verification efficiently for large-scale application is very challenging, considering the expensive computational cost of full location verification.

In this paper, we propose to address large-scale similar song retrieval by using effective music features and location coding strategies. We define two songs as similar songs when they share some similar beat-chroma patches with the same or very similar location relationship. Our approach is based on the Bag-of-Audio-Words model. To improve the discrimination of audio words, we utilize the beat-aligned chroma feature for codebook construction. All western music can be represented through a set of 12 pitches or semitones. Beat-aligned chroma features record the intensity associated with each of the 12 semitones for a single octave during a defined time frame. We measured the intensity of the 12 semitones in the time frame of a beat. To verify the matched local parts of two songs, we apply location coding scheme to encode the relative positions of local features in songs as location maps. Then through location verification based on location maps, the false matches of local features can be removed effectively and efficiently, resulting in good retrieval precision.

The contribution of this paper can be summarized as follows: 1) we apply beat-chroma patterns to build descriptive audio codebook for large scale similar song retrieval; 2) we utilize location coding method to encode the relative location relationships among audio words into location maps; 3) we apply a location verification algorithm to remove false matches based on location maps for similar song search.

The rest of the paper is organized as follows. In Section 2, related work is introduced. Then, our approach is illustrated in Section 3. In Section 4, some preliminary experimental results are provided. Finally, we make the conclusion in Section 5.

2 RELATED WORK

The codebook approach for large scale music retrieval is proposed in (Seyerlehner et al., 2008). By using vector quantization, a large set of local spectral features are divided into groups. Each group corresponds to a sub-space in the feature space, and is represented by its center, which is called an audio word. All audio words constitute an audio codebook. With local features quantized to audio words, song representation is very compact. And by inverted-file index, all the songs can be efficiently indexed as bag-of-audio-words, achieving fast search response.

Recently, beat-chroma feature becomes a popular and useful local feature for music retrieval and classification. By learning codebook from millions of beat-chroma features, the common patterns in beat-synchronous chromagrams of all the songs can be obtained (Bertin-Mahieux et al., 2010). Each individual codeword consists of short beat-chroma patches of between 1 and 8 beats, optionally aligned to bar boundaries (Bertin-Mahieux et al., 2010). This approach dug the deeper common patterns underlying different pieces of songs than the previous "shingles" (Casey and Slaney, 2007).

However, beat-chroma feature quantization reduces the discriminative power of local descriptors. Different beat-chroma features may be quantized to the same audio word and cannot be distinguished from each other. On the other hand, with audio word ambiguity, descriptors from local patches of different semantics may also be very similar to each other. Such quantization error and audio word ambiguity will cause many false matches of local features between songs and therefore decrease retrieval precision and recall.

To reduce the quantization error, one solution is to utilize location information of local features in songs to improve retrieval precision. Many geometric verification approaches (Wu et al., 2009; Zhou et al., 2010) have been proposed for image retrieval. Among them, spatial coding approach (Zhou et al., 2010) is an efficient global geometric-verification method proposed to verify spatial consistency of features in the entire image, which relies on visual words distribution in two dimensional images. Motivated by the spatial coding algorithm (Zhou et al., 2010), we apply its simpler version to encode the relative location relationships

among audio words in a song into one dimensional location map.

In this paper, we focus on large scale similar song search, including remix songs. Despite research in the detection of cover songs (Ellis and Poliner, 2007) and songs with partial missing data (Bertin-Mahieux et al., 2011), little work yet has focused on the identification of remix songs. This identification can be difficult since many times remix versions of songs introduce a large amount of noise to the original song. Our motivation is to build an effective codebook with an efficient global location verification scheme, which can achieve both accuracy and efficiency of real-time response.

3 OUR APPROACH

3.1 Feature Extraction

In our approach, we adopt beat-chroma features (Bertin-Mahieux et al., 2010) for song representation. In order to extract the beat-chroma feature, the Echo Nest analyse API (http://the.echonest.com/) is used. For any song uploaded to Echo Nest API, it returns a chroma vector (length 12) for every music segment and a segmentation of the song into beats and bars. Beats may span or subdivide segments; bars span multiple beats. Averaging the per segment chroma over beat times results in a beat-synchronous chroma feature representation (Bertin-Mahieux et al., 2010).

Once the chroma features have been extracted, they are then grouped together based on the time signature. Grouping the features together based on their time signature is essentially to develop comparable features since beats are commonly grouped into measures dictated by a time signature. For example, if the song is 4/4 time, there would be 4 chroma-beat features to a group to compose a chroma-beat patch. The time signature of the song is also determined using the Echo Nest analyse API.

3.2 Codebook Construction

Gathering good musical features for codebook generation requires having access to a large set of songs. In this paper, we use a large song dataset (LSD) (Bertin-Mahieux et al., 2010), consisting of 43.3 K songs with 3.7 million extracted non-silent beat-chroma features.

Based on these million-scale features, we construct a codebook by using hierarchy *K*-means clustering (Nister and Stewenius, 2006) with *L* levels. The resulting cluster centroids are regarded as audio words. The standard *K*-means algorithm performs a NP hard problem to which the upper bound is $O(K^n)$ (Arthur and Vassilvitskii, 2006). To improve runtime, we use the *k*-means++ algorithm which uses careful seeding to improve the performance to $O(\log k)$. The final codewords used to represent the dataset are at the bottom of the tree structure. This tree structure greatly improves the efficiency of our indexing, since it can be traversed in logarithmic time. The entire dataset can then be represented by the K^L histograms.

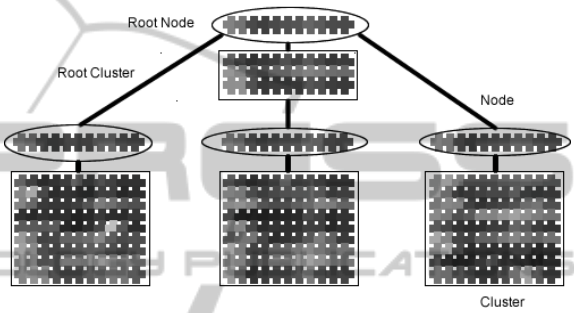


Figure 1: Feature Dendrogram.

In our experiments, we set *K* to be 10 and vary our *L* to produce different amounts of codewords. The resulting dendrogram is then written to an XML document and stored on disk. Since we will not be transforming our dendrogram into an index online, we would not be losing efficiency in our retrieval system by writing the dendrogram to a disk. An example of the feature dendrogram can be seen in Figure 2 and 3.

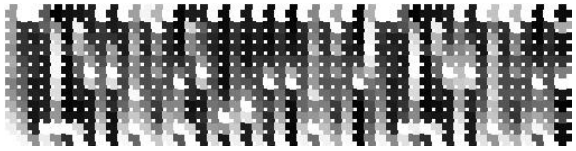


Figure 2: Original Song beat-aligned chroma.

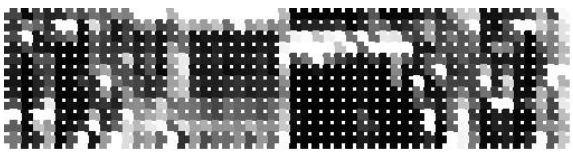


Figure 3: Remix Song beat-aligned chroma.

3.3 Feature Quantization

To build a large scale song retrieval system, we need to quantize all local beat-chroma features into audio

words. Assuming that the similar songs share similar beat-chroma patterns and locations in both the target and query songs, a pair of truely matched features should share similar beat-chroma values.

All the 3.7 million beat-chroma features are quantized to audio words based on the built vocabulary tree (codebook). During the quantization stage, a novel feature will be assigned to the index of the closest audio word (centroid). With feature quantization, any two features from two different songs quantized to the same audio word will be considered as a local match across two songs.

3.4 Location Coding and Verification

After feature quantization, matching pairs of local features between two songs can be obtained. However, the matching results are usually polluted by some false matches. To efficiently filter those false matches without sacrificing accuracy, we apply location coding scheme for location verification.

Location coding (LC) encodes the location context between each pair of beat-chroma features in a song. In LC, with each beat-chroma feature as reference origin, the song is divided into two parts (Figure 4). A location coding map, called *L*-map, is constructed by checking whether other features appear before or after this feature in the song. For instance, given a song S with M features $\{f_i\}$, $(i=1,2,\ldots,M)$, its *L*-map is defined as follows:

$$Lmap(i,j)=\begin{cases}0 & \text{if } v_j \text{ appears before } v_i \text{ in the song}\\ 1 & \text{otherwise}\end{cases}$$

$$(1\le i,j\le M) \qquad (1)$$

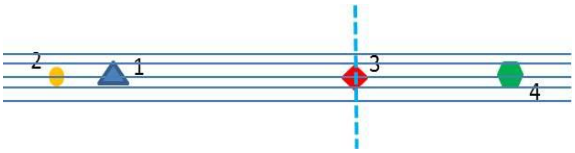


Figure 4: An illustration of location coding for song features.

Figure 4 gives a toy example of song division with the key point of feature 3 as reference point. The resulting *L*-map is:

$$Lmap=\begin{Bmatrix}1\,0\;1\,1\\ 1\,1\;1\,1\\ 0\,0\;1\,1\\ 0\,0\;0\,1\end{Bmatrix}$$

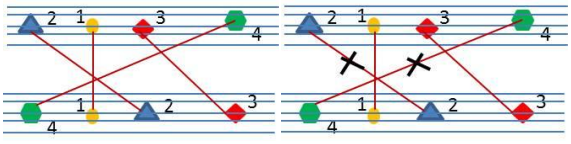
In row i, feature v_i is selected as the origin, and the song is divided into two parts (Figure 4). Then row i records other features' location relationships with feature v_i in the song. For example, $Lmap(1,2)=0$ means feature v_2 appears before feature v_1 in the song. Therefore, one bit either 0 or 1 can encode the relative location position of one feature to another in one coordinate.

Given that a query song S_q and a matched song S_m are found to share N pairs of matched features through beat-chroma quantization. Then the corresponding sub-location maps of these matched features of both S_q and S_m can be generated and denoted as L_q and L_m by Eq. (1). After that, we can compare these location maps to remove false matches.

Since the *L*-map is binary, for efficient comparison, we can perform logical Exclusive-OR operation on L_q and L_m, $V(i,j)=L_q(i,j)\oplus L_m(i,j)$. Ideally, if all N matches are true, V will be zero for all their entries. If some false matches exist, the entries of these false matches on L_q and L_m may be inconsistent. Those inconsistencies will cause the corresponding exclusive-OR result to be 1.

$$\text{Denote } S(i)=\sum_{j=1}^{N}V(i,j) \quad (i=1\sim N) \qquad (2)$$

If for some i with $S(i)>0$ we define $i^*=\arg\max_i S(i)$, then the i^*th pair will be most likely to be a false match and should be removed. Consequently, we can iteratively remove such mismatching pairs, until the maximal values of S is zero.



(a) (b)

Figure 5: An illustration of location verification on a relevant song pair. (a) Initial matched feature pairs after quantization; (b) False matches detected by location verification.

Figure 5 shows an example of the location verification with location coding on a relevant song pair. Initially, the relevant song pair has four matches of local features (Fig.5 (a)). After location

verification via location coding, two false matches are identified and removed, while two true matches are satisfactorily kept (Fig.5 (b)). With those false matches removed, the similarity between songs can be more accurately defined and that will benefit retrieval accuracy. The philosophy behind the effectiveness of our location verification approach is that, the probability of two irrelevant songs sharing many location consistent audio words is very low.

3.5 Indexing

Indexing of songs with the codewords is essential to improve the runtime of similar songs matching. We use an inverted file structure to index songs. As illustrated in Figure 6, each audio word is followed by a list of indexed features that are quantized to the audio word. Each indexed feature records the ID of the song where the audio word appears. Besides, for each indexed feature, its location information in that song is also recorded, which will be used for generating location coding maps for retrieval. This index is then written to file and will be loaded into main memory at query time.

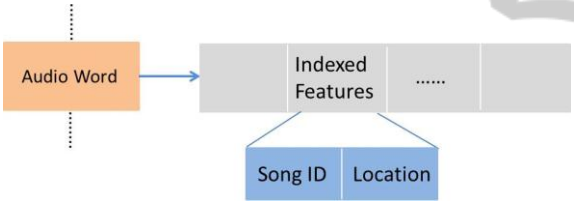


Figure 6: Inverted file structure for index.

4 EXPERIMENTS

4.1 Experimental Setup

We build our basic dataset by using the large song dataset (LSD) (Bertin-Mahieux et al., 2010), consisting of 43.3 K songs with 3.7 million extracted non-silent beat-chroma features. For evaluation of similar songs retrieval, we select 20 songs from the US Billboard Hot 100 charts, which encompass 12 different genres. We compile a collection of 72 similar songs to these 20 songs through the Internet. These 72 similar songs are then appended to the LSD song dataset to form our similar song dataset with ground truth labels.

We use beat-chroma feature for song representation. The Echo Nest API is used to extract features for the query songs and similar songs. The similar songs have features extracted prior to query time. Only the query song has features extracted online.

4.2 Retrieving Similar Song

Given a query song, the beat-chroma features are extracted from the query song online by using the Echo Nest API. Then based on the built codebook, all the features on the query song are quantized to their closest audio words using cosine similarity (Eq. (3)). The most similar song is the song which contains the largest number of common codewords with the query song.

$$s(q_1, q_2) = \arccos\left(\frac{\vec{q_1} * \vec{q_2}}{\|\vec{d_1}\|\|\vec{d_2}\|}\right) \quad (3)$$

To find the most similar song, for each codeword on the query song, we use the inverted feature index to find the songs which have the same codeword. These songs receive a vote when they have a matching feature in common with the query song. Once we have gone through all the codewords in the query song, we then return the rank of similar songs in the order of the number of votes received.

4.3 Preliminary Experimental Results

In our preliminary experiments, we first evaluate our codebook's accuracy based on the average precision (AP), which is widely used in information retrieval.

The AP averages the precision values obtained when each relevant similar song occurs in our retrieval results. The AP of the top-T ranked songs AP@T is calculated as follows:

$$AP@T = \frac{\sum_{r=1}^{N}\left(P(r) * rel(r)\right)}{relevant\ \ partial\ duplicate\ songs} \quad (4)$$

In our experiments, we consider several different T values.

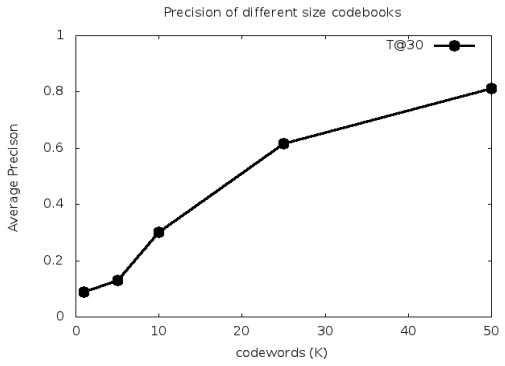


Figure 7: Precision for different sized codebooks.

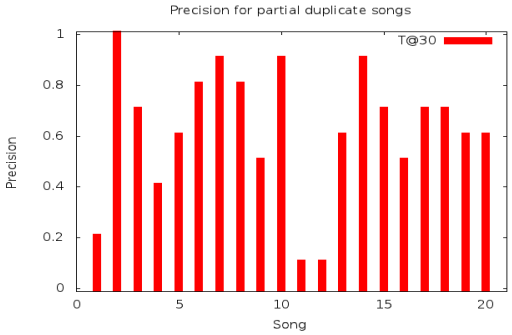


Figure 8: Precision for each query song.

We compile several codebooks of different sizes. As indicated in (Van Gemert et al., 2010), having too small of a codebook does not discriminate well between feature categories or help us build an understanding of the most important features in a dataset. For this reason, current state-of-the-art codebook approaches typically contain several thousands of codeword (Marszalek et al., 2007). We evaluate several codebook sizes for our large song dataset and observe how it affects query performance by obtaining the average precision and recall for each.

From Figure 7, we find that as the size of the codebook becomes larger, the average precision also increases. This is consistent with the observation in (Van Gemert et al., 2010), that having too small of a codebook does not allow us to discriminate well between feature categories. As shown in Figure 8, the retrieval of particular songs yields higher precision than others. In the queries yielding low levels of precision, it is difficult to accurately extract the beat-aligned chroma features from these songs.

Based on the preliminary experiments, in the next step, we will further compare the location coding method with the traditional bag-of-audio-words model (baseline) and other state-of-the-art location verification approaches on search precision, recall, query time, and computational cost.

5 CONCLUSIONS

In this paper, we apply beat-chroma patterns to build a descriptive audio codebook for large-scale similar song retrieval. A location coding scheme is used for audio words match verification. The location coding efficiently encodes the relative locations among features in a song and effectively discovers false feature matches between songs. Our approach shows some success however, on remix songs that have large amounts of bass or drum beats layered over them. Our approach is also novel since we attempt to extract features from songs covering a large variety of musical genres and we do not focus on one particular genre. The hierarchical *K*-means clustering algorithm used performs well given the size of the large song dataset.

In the future, a more careful analysis of a way to remove noise from remix songs will be performed, so that the beat-aligned chroma features can be more accurately assessed. More comprehensive experiments on comparing the proposed approach with the bag-of-audio-words method and other state-of-the-art location verification approaches will be conducted. Their retrieval accuracy, storage cost, and query speed will be fully analysed.

ACKNOWLEDGEMENTS

This work is supported by NSF REU grant 1062439, NSF-CISE CRI Planning Grant 1058724, Research Enhancement Program of Texas State University, and DoD HBCU/MI grant W911NF-12-1-0057.

REFERENCES

Arthur, D. and Vassilvitskii, S., 2006. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pages 144-153, New York, NY, USA.

Bertin-Mahieux,T., Grindlay, G., Weiss, R., and Ellis, D., 2011. Evaluating music sequence models through missing data. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Bertin-Mahieux,T., Weiss, R., and Ellis, D., 2010, Clustering beat-chroma patterns in a large music database. *11th International Conference on Music Information Retrieval (ISMIR)*.

Casey, M., and Slaney, M., 2007. Fast recognition of remixed music audio. In *Proceedings of ICASSP*.

Ellis, D. and Poliner, G., 2007. Identifying "cover songs" with chroma features and dynamic programming beat tracking. *Proceedings of ICASSP*.

Maddage, N. C., Xu, C., Kankanhalli, M. S., and Shao, X., 2004. Content-based music structure analysis with applications to music semantics understanding. *ACM Multimedia*, pages 112-119, New York, NY, USA.

Marszalek, M., Schmid,C., Harzallah,H., and Van De Weijer, J., 2007. Learning object representations for visual object class recognition. *Visual Recognition Challenge workshop, in conjunction with ICCV*.

Nister, D. and Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR). New York, pages 2161-2168.

Seyerlehner, K., Widmer, G., and Knees, P., 2008. Frame level audio similarity-a codebook approach. *Proceddings of the 11th International Conference on Digital Audio Effects*.

Van Gemert, J. C., Veenman, C. J., Smeulders, A. W. M., and Geusebroek, J.M., 2010. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271-1283.

Vedaldi, A. and Fulkerson, B., 2010. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the International Conference on Multimedia*, pages 1469-1472, New York. USA.

Wu, Z., Ke, Q., Isard, M., and Sun, J., 2009. Bundling features for large scale partial-d*uplicate* web image search. In *Proceedings of CVPR*.

Zhou, W., Lu, Y., Li, H., Song, Y., Tian, Q., 2010. Spatial coding for large scale partial-duplicate web image search. *ACM Multimedia*, Florence, Italy.

http://the.echonest.com/platform/