Local Summarization and Multi-Level LSH for Retrieving Multi-Variant Audio Tracks

Yi Yu
Department of Computer Science, New Jersey Institute of Technology
University Heights Newark NJ 07102
yuyi@njit.edu

Michel Crucianu
Vertigo - CEDRIC, Conservatoire National des Arts et Métiers
292 rue St Martin, 75141 Paris Cedex 03, France
michel.crucianu@cnam.fr

Vincent Oria
Department of Computer Science, New Jersey Institute of Technology
University Heights Newark NJ 07102
oria@njit.edu

Lei Chen
Department of Computer Science, HKUST
Clear Water Bay
Kowloon Hong Kong
leichen@cs.ust.hk

ABSTRACT

In this paper we study the problem of detecting and grouping multi-variant audio tracks in large audio datasets. To address this issue, a fast and reliable retrieval method is necessary. But reliability requires elaborate representations of audio content, which challenges fast retrieval by similarity from a large audio database. To find a better trade-off between retrieval quality and efficiency, we put forward an approach relying on local summarization and multi-level Locality-Sensitive Hashing (LSH). More precisely, each audio track is divided into multiple Continuously Correlated Periods (CCP) of variable length according to spectral similarity. The description for each CCP is calculated based on its Weighted Mean Chroma (WMC). A track is thus represented as a sequence of WMCs. Then, an adapted two-level LSH is employed for efficiently delineating a narrow relevant search region.The "coarse" hashing level restricts search to items having a non-negligible similarity to the query. The subsequent, "refined" level only returns items showing a much higher similarity. Experimental evaluations performed on a real multi-variant audio dataset confirm that our approach supports fast and reliable retrieval of audio track variants.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.5.5 [**Information Systems**]: Sound and Music Computing

MM'09, October 19–24, 2009, Beijing, China.

General Terms

Algorithms, Performance, Experimentation

Keywords

Multi-variant musical audio search, local audio summarization, multi-level LSH

1. INTRODUCTION

Musical audio content represents a significant share of the user-generated content on the Web. In many cases (for example `www.secondhandsongs.com` or `www.midomi.com`), such content corresponds to popular songs that are interpreted, recorded and uploaded by different people, in various moods and sometimes from different countries. The resulting collections of *multi-variant audio tracks* are found interesting by many users. For example, statistics[1] show that a site like `secondhandsongs.com` is visited by approximately 12,400 people every month. Archives storing large audio databases would like to provide relevant similarity-based retrieval services to the users. These archives would also appreciate being able to structure their content by grouping together the different interpretations of a same song. Unfortunately, to find the variants of a song, one cannot rely on textual metadata since the annotations are often in different languages or simply missing. Furthermore, the direct comparison of audio-based content descriptions is challenging because there may be significant differences between versions and also because of the high cost of comparing with many long sequences of high-dimensional audio descriptions.

The effective and efficient detection of multi-variant audio tracks is thus a motivating topic in Music Information Retrieval (MIR), related to musical audio signal processing [1, 2], audio content indexing [3, 4], sequence matching [1, 4, 5], music retrieval evaluation [6], etc. It has already attracted significant research interest, see e.g. [1, 2, 3, 4]. The previous proposals rely on comparing either sequences of audio

[1] `www.quantcast.com/secondhandsongs.com`

frame descriptions (feature sequences) [1, 2] or on extracting very compact descriptions of entire audio sequences [6, 7]. The use of feature sequences (denoted by $\{r_{i,j}\}$) leads to more accurate retrieval but does not scale well to large audio databases because matching long sequences is expensive. Alternatively, very compact descriptions of entire audio sequences (where an entire feature sequence $\{r_{i,j}\}$ is summarized by a single vector V_i) support good scalability but the accuracy of the descriptions (and, consequently, of retrieval) is limited. An important challenge in detecting and grouping multi-variant audio tracks is to achieve a good balance between accuracy and efficiency over large audio datasets.

To address this challenge, we put forward in this paper a *Local* Summarization (LS) method and a *multi-level* Locality Sensitive Hashing (LSH) scheme. Based on a prior study of audio features that can support the detection of multi-variant audio tracks, each audio track is divided into Continuously Correlated Periods (CCP). A Weighted Mean Chroma (WMC) description is computed for each CCP and can be seen as a local summary. The sequence of WMCs describing the consecutive CCPs is a concise yet relatively precise representation of the entire audio sequence. It helps improve the accuracy of variant track retrieval without requiring expensive computation. We further suggest a two-level LSH scheme for efficiently delineating a narrow relevant search region. At the first level, a "coarse" hashing is performed in order to restrict search to items having a non-negligible similarity to the query item. To find those items that are highly similar to the query, a subsequent "refinement" hashing is used. This significantly accelerates retrieval by similarity of multi-variant audio tracks, while providing good recall and precision. The proposed method can first be employed for directly answering user queries sent to the server, by returning the top k most similar tracks, among which the variants should rank well. The same method can serve for grouping together the different variants of the audio tracks in the database, either offline for the entire database or online if the answers to some query contain variants of one or several audio tracks. To show that the method is relevant for real-life applications, we run our algorithms over a large musical audio dataset with real multi-variant queries recorded by different users. The evaluation results show that the proposed method has a better tradeoff between retrieval speed and quality than other methods, especially when queries are shorter than their covers in the database.

This paper is organized as follows. We report the research background and the related work in section 2 and describe the structural analysis based on Chroma in section 3. In section 4, we present the main components of our approach, first introducing the idea of local summarization relying on spectral similarity, and then explaining how to build an adapted locality sensitive mapping exploiting Chroma energy in order to assign the hash values. An analysis of the algorithm is also provided. Performance evaluation of the proposed approach is conducted over a large dataset. The experimental setup and analysis of the results are given in section 5. We conclude with a summary of our proposals and findings.

2. RELATED WORK

In MIR, query-by-content consists in searching the database using the audio itself as query. This can be performed by directly looking for audio tracks whose content descriptions are similar to the description of the query [2, 3, 4], according to some relevant similarity measures. It is also possible to first map the query music to some related category (e.g. to genre [8] or emotion [5]) based on its audio content, and then return the tracks that belong to the selected categories. For musical content, similarity can be defined in many different ways depending on the search intent, personal opinion or interest, cultural background, etc.

The detection of multi-variant musical audio tracks is considered in previous work (see [1, 2, 3, 4]) as a sub-topic of query-by-content in MIR. It is an interesting and motivating subject, especially with more and more unknown audio recordings being uploaded to User Generated Content (UGC) websites. More specifically, the systems that were proposed take an audio track as query, perform search by similarity and return the resulting tracks in a list ordered by decreasing similarity to the query. In this domain, the main research issues are about finding the appropriate representation of music content and the organization of audio track descriptions in order to support fast and accurate retrieval. Regarding the first issue, the aim is to improve the accuracy of multi-variant audio track detection and the different proposals rely on pitch [9, 10], Mel-Frequency Cepstral Coefficients (MFCC) [11, 12, 13] or Chroma [2, 14]. With regard to the latter research issue, the goal is to accelerate the retrieval by similarity and the existing proposals include tree structures [7, 15, 16], other hierarchical structures [17], LSH [4, 11, 18], Exact Euclidean LSH (E^2LSH) [3, 4] and other variants of LSH [6, 11]. It is however clear that the two research issues are not independent, since more accurate detection requires more elaborate representations of audio content, with a negative impact on scalability.

Some research has focused on the extraction of better music features and performs a complete audio sequence comparison on the entire dataset. Sequences of Chroma features [2, 14] were successfully used in matching multi-variant music sequences. They provide good retrieval accuracy, but require rather expensive sequence comparisons. Log-Frequency Cepstral Coefficients (LFCC) and chromagram features [3] were also successfully used for nearest-neighbor music searches. LSH was applied in many cases in order to accelerate the retrieval of similar sequences from large repositories. Shingles were created by concatenating consecutive frames and used as higher-dimensional features. Then E^2LSH was adopted to retrieve candidates that are similar to the query [3]. In [4], with LSH or E^2LSH to support similarity-based retrieval, the resulting Short Time Fourier Transform (STFT) features are reorganized into partial sequences and compared with the query by either Dynamic Programming (DP) or Sparse DP (SDP). In [11], MFCC are employed and multi-probe LSH is introduced in order to investigate multiple buckets that are likely to contain items similar to the query. In [15], the features that are based on the Discrete Fourier Transform (DFT) are grouped by Minimum Bounding Rectangles (MBR) and indexed using a spatial access method. Yang [18] used random sub-sets of STFT features to compute hash values for parallel LSH hash instances. With a query as input, the relevant features are matched using hash tables. For bucket conflict resolution, a Hough transform is performed on these matching pairs to detect the similarity between the query and each reference song by linearity filtering.

Other research has rather considered extracting a single compact vector feature from each entire audio sequence and

then comparing the resulting vectors. In [7], a composite feature tree (using e.g. timbre, rhythm, pitch) was proposed to facilitate the search for the k nearest neighbors (kNN). A summary is generated from a feature sequence, by using multivariable regression and Principal Component Analysis (PCA). In [6], weights are assigned to frequently employed features like MFCC, Chroma, Mel-magnitudes, based on a principle of spectral similarity invariance. A long audio feature sequence is summarized as a compact single Feature Union (FU). Then SoftLSH is employed for better locating the relevant search region.

From the existing work it can be concluded that a feature sequence as representation for a musical audio track has a high description accuracy but poor conciseness. At the opposite, a global feature summary is a very compact representation but its accuracy is comparatively low. In this work, we focus on the generation of *local summaries* in order to find a better tradeoff between accuracy and conciseness in the representation of musical audio sequences for multi-variant track detection. The coefficient of spectral correlation between adjacent audio frames is computed and compared against a pre-determined threshold to segment a musical audio track into Continuously Correlated Periods (CCP). We summarize each CCP by its Weighted Mean Chroma (WMC) features.

The WMC descriptions are then mapped to hash keys with the help of locality sensitive hash functions. To delineate a narrow relevant search region, an adapted two-level LSH scheme is proposed, based on the relative energy of the 12 semi-tones in the WMC. The positions of several major bins (i.e. bins having high energy) are used to determine the hash keys identifying the buckets in the first-level hash table. The WMC features in each bucket have a non-negligible similarity. A specific quantization of the actual energy values of the major bins allows dividing each bucket in the first-level hash table into non-overlapping blocks that can be regarded as a second-level hash table. The WMC features in each block have a high similarity. Queries are answered by multi-probing several similar blocks, which results in a good recall with almost no negative impact on precision. The second-level hash tables can be refined in order to further speed up retrieval and reduce the amount of inactive storage space.

3. MUSIC REPRESENTATION

Since the audio collections are very rich datasets, a major task in music signal processing is to extract a representative audio profile that depicts the acoustic-related music content of each song. Appropriate features should allow to distinguish among songs and, at the same time, be insensitive to the differences between the variants of a *same* song. Moreover, compact representations of the musical audio information should support more efficient retrieval and require less additional storage. In this section we review some typical music representations and present the reasons for choosing Chroma as the base feature.

3.1 Representation of Music Signals

Unlike other kinds of audio, music has strong descriptive composition. Different music pieces usually have different scores (sheet music). The score represents the most concise description of a song. However, translating an audio signal to a score is quite difficult for polyphonic songs. Most representations are based on some audio features and can be

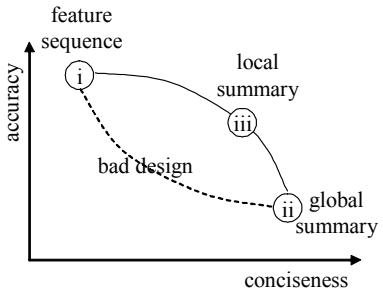

Figure 1: Accuracy-conciseness tradeoff.

categorized as follows:

(i) *Sequence of features.* Music signals have regular spectral structures that are highlighted by their scores. This is why spectral features, such as MFCC [11], STFT [18], pitch [9], and Chroma [14], have been widely exploited in music retrieval. A music signal is only stable for a short period of time. Hence, a signal is often divided into short frames, from which some features are extracted and represented as multidimensional vectors. Although the sequence of features retains a large share of the information present in a music signal, using it directly to retrieve music can be expensive due to its high temporal length.

(ii) *Global summarization.* One way to reduce the computation cost is to summarize the audio signals. In [6] audio signals are represented by statistics of a set of frequently used features. Although summarization effectively reduces the volume of data, a large part of the temporal information is lost and the accuracy decreases when the summaries are used for retrieval. For this scheme to work well, one has to make sure that a query has almost the same statistics as its relevant tracks in the dataset.

(iii) *Local summarization* is the focus of this paper. A music signal is divided into multiple segments so that the statistics of each segment remain almost unchanged along the segment and can thus be used as a local summary with little information loss.

As shown in Figure 1, (i) and (ii) are two extremes of music signal representation: (i) has the highest accuracy but a large amount of redundancy and (ii) is very concise but also loses significant information since it exclusively relies on global statistics. It is obvious that a tradeoff is necessary. In the following we adopt *local* summarization, which is adequate since the musical audio signal is short-term stationary. It was reported in [19] that adjacent frames corresponding to the same note are highly correlated. In this work, the music signal is divided into multiple stable periods, each generating a local summary. This approach is also very efficient since the redundancy is significantly reduced.

3.2 Spectral Properties of Chroma

Chroma plays an important role in music perception [20], and is often used in content-based musical information recognition and detection [1, 2, 6, 14]. Since Chroma features only capture tonal information, they are invariant to some differences among multi-variant audio tracks. In the following we further investigate some spectral properties of Chroma to give the reasons why we can use the statistical properties of Chroma energy distribution.

The spectrum of music signal is structured, showing a cer-

tain number of harmonics. The entire frequency band of music signal can be divided into 88 sub-bands in such a way that the central frequency of each sub-band is $2^{1/12}$ times of its previous one. Each sub-band corresponds to a note/pitch. Frequencies that are one octave apart (frequency ratio 2:1) represent harmonics and constitute a frequency family. As a result, there are 12 distinct note families. When pitch is used as the feature, the note with maximal energy is extracted. In such cases, harmonics are not discriminative and it is not easy to determine which of a frequency family reflects the real pitch. Therefore, sub-harmonic summation is exploited to explicitly distinguish a note from its harmonics.

Notes belonging to the same family are perceived as being similar to each other. Hence, it is unnecessary to distinguish harmonics. Chroma is an interesting and powerful representation for musical audio, in which the energy of each frequency family is calculated separately. It is a 12-dimension vector corresponding to the 12 distinct frequency family. Since, in music, frequencies exactly one octave apart are perceived as particularly similar, knowing the distribution of Chroma, even without the absolute frequency, can give useful musical information about the audio and may even reveal a perceptual musical similarity that is not apparent in the original spectrum. Please refer to [2] for details regarding Chroma features.

By definition, each bin of Chroma represents the total energy of the corresponding frequency family. It can be calculated from the power spectrum by applying 12 discrete windows. MFCC also considers the human auditory system, but Chroma is different: MFCC focuses on the continuous frequency band while Chroma focuses on discrete harmonics.

4. PROPOSED APPROACH

Content-based musical audio retrieval involves both music signal processing and audio content indexing. In this section we present a multi-level LSH scheme based on a local summarization method. They are both designed to satisfy the requirements of scalable retrieval of multi-variant audio tracks. We describe the representation of an audio signal with local summaries, the computation of WMC features based on the Chroma spectral energy distribution, the conversion of WMC sequences into hash values, the organization of local summaries of audio tracks with the two-level LSH structure, and the overall retrieval process.

4.1 Local Summarization

As mentioned earlier (see also [19]), spectral features of adjacent frames corresponding to the same note are highly correlated. Spectral similarity is used as the main metric to determine continuously correlated periods (CCP). Let c_i be the Chroma feature of the i^{th} frame of a song. The correlation between two Chroma features c_i and c_j is calculated according to Eq.(1), where $< \cdot, \cdot >$ is the inner product between two vectors.

$$\rho = \frac{< c_i, c_j >}{\sqrt{< c_i, c_i >< c_j, c_j >}} \quad (1)$$

Figure 2 shows an example of correlations between adjacent frames. The piece of audio signal was divided into 19 frames. The numbers on top of the figure are correlation coefficients between the Chroma features of adjacent frames. By comparing these correlation coefficients against a pre-determined threshold ρ_{th}, CCPs can be found. For

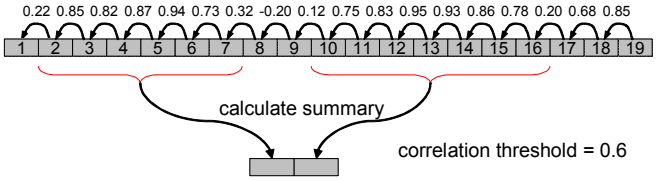

Figure 2: Local summarization.

example, frames 2 to 7 form the first CCP and frames 10 to 16 form the second CCP. A CCP represents the stable period of a note. Between two CCPs there are also frames, such as 8 and 9, having low correlation with adjacent frames. These frames represent unstable (transitory) periods and are neglected in our summarized representation.

Each CCP j has a run length L_j. Frames in the same CCP correspond to the same note and have almost the same spectral structure. Let the k^{th} frame in CCP j be c_{jk}. Instead of keeping all the frames in a CCP, their common information is extracted as a prototype. The spectrum is the most stable in the middle of a CCP. Therefore, the features of all the frames in a CCP are weighted by a triangle window of length L_j, as shown in Eqs. (2-3). This produces a weighted mean Chroma (WMC) feature, regarded as the summary of this CCP. By segmenting the signal into CCPs using the similarity threshold ρ_{th} and keeping only the WMC for each CCP, the resulting feature sequence is compact while keeping significant temporal information of the audio track. The impact of local summarization is discussed in section 5.2.

$$T_{jk} = \begin{cases} k/L_j, & k \leq L_j/2 \\ (L_j - k)/L_j, & k > L_j/2 \end{cases} \quad (2)$$

$$WMC_j = \sum\nolimits_k T_{jk} \cdot c_{jk} \quad (3)$$

4.2 Quantization of WMC

Each audio track is represented by a sequence of WMC after summarizing the original Chroma features of the song. To organize the WMCs in the database we define an adapted LSH scheme. Hash values are calculated from the quantized WMC. To retain the perceptual similarity in the quantization stage, an investigation of some characteristics of the WMC features is presented below.

Each Chroma feature has $V = 12$ dimensions (or 12 bins). These bins represent energies of semi-tones, or frequency families. It is known that in monophonic song there is a primary note, while in polyphonic songs there may be several notes simultaneously initiated. Anyway, the number of simultaneous notes is usually limited and the energy of each Chroma feature concentrates on few major bins.

We study the energy distribution over different bins in each Chroma feature. The dimensions in each Chroma feature are sorted in the descending order of their energy and these bins are called 1^{st} major bin, 2^{nd} major bin, and so on. The energy of first k (1^{st}, 2^{nd}, $\cdots$, k^{th}) major bins is calculated and normalized by the total energy of the Chroma feature. Figure 3 shows the cumulative density function of these normalized energies when $k = 1, 2, \cdots, 6$, respectively. This allows to see that the energy of Chroma features concentrates on few bins, with the $L = 4$ major ones representing more than 80% of the energy. The energies of the other bins are too low to have a significant contribution to the

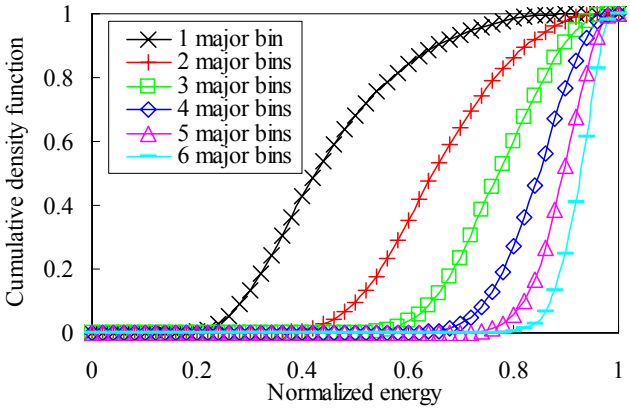

Figure 3: CDF of normalized energy.

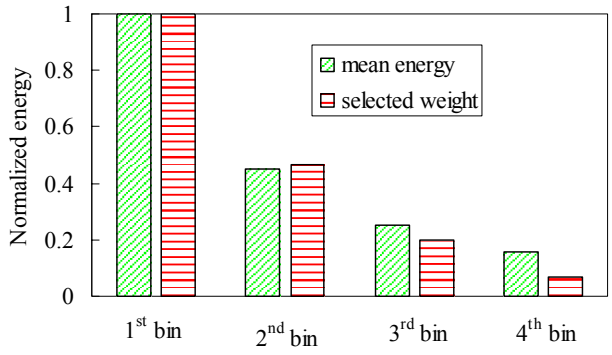

Figure 4: Mean energy and weights for major bins of Chroma.

similarity and these bins can be safely neglected. Therefore only the first four major bins ($L = 4$) are kept.

The major bins of Chroma should also be quantized in order to calculate integer hash values. A major property of audio spectrum is that the perceptual similarity is determined not by the absolute strength of the spectrum, but rather by the relative strength. In other words, the order of major bins arbitrates the perceptual similarity. Therefore, instead of performing accurate quantization, we take the relative strength into account and decide to assign weights to the selected L major bins. The actual weight depends on the relative strength of each major bin, as shown in Figure 4. The energies of 1^{st}, 2^{nd}, 3^{rd}, 4^{th} major bins are averaged over all Chroma features and normalized so that the energy of the 1^{st} major bin equals 1. The weights are selected to fit the normalized energy and, at the same time, reflect well the relative strength. The chosen weights are $2^{L-i+1} - 1$ for the i^{th} major bin, i.e. 15, 7, 3 and 1 for $L = 4$. To each of the other $V - L$ bins, a weight of 0 is assigned.

4.3 Two-Level LSH Structure

LSH is a hash-based method employed in approximate search and retrieval schemes [3, 4, 6, 11, 21, 22] to find all the items similar to a query. More specifically, features are extracted from items and regarded as similar to one another if they map to the same hash values. Locality sensitivity ensures that similar items collide in the same bucket with a high probability. But not-so-similar features can also share the same bucket. To improve the precision, a post-filtering is required. The filtering stage takes much time in LSH schemes and depends on the percentage of non-similar items in the same bucket. In addition, with a single hash instance, recall values can relatively be low. Usually, several parallel hash instances are used to improve recall values. This further increases computation costs. In most LSH schemes, a single-level structure is used and it is difficult to find a trade-off between retrieval quality and computation cost.

In the following, we address this problem by designing a two-level hash structure in the light of Chroma energy distribution. Dividing the hash tables into two levels facilitates the design of LSH functions. In the first level hash table, each bucket contains features with non-negligible similarity. Each bucket is further divided into blocks and forms a second-level hash table. Features in the same block have higher similarity. To meet the different similarity requirement, different LSH functions are used at the two levels.

Organizing the music features in the database via LSH requires computing the hash values from the features. Although the random selection of sub-dimensions or the application of random linear functions to the WMC generates hash values, we do this in a different way, better adapted to the nature of the WMC features. In the previous section, we showed that from the i^{th} WMC, L major bins $P_i =< p_{i1}, p_{i2}, \cdots, p_{iL} >$ are assigned non-zero weights $H_i =< h_{i1}, h_{i2}, \cdots, h_{iL} >$. Two WMCs with some common non-zero positions share some frequencies and are similar to some extent. If the assigned weights are also the same, the similarity degree increases. Therefore, the hash values in the first hash table are calculated from the *positions* of the L major bins. Hash values in the second hash table are calculated from the *weights* of the L major bins. Because two WMCs may only share part of the non-zero frequencies, a subset of P_i (with C positions, $C < L$) determines a bucket in the first hash instance.

Similar to the parallel hash instances in general LSH schemes, some redundancy is necessary to ensure a relatively high recall in the proposed scheme. Let P_{ij} be the j^{th} subset of the position set P_i and H_{ij} the corresponding subset of the weight set H_i. In this way, each WMC appears in several buckets in a first level hash instance and the number of occurrences depends on the number of subsets P_{ij}. In the second level hash instance, there is no overlapping between blocks. Each block is associated to a subset H_{ij}.

Figure 5 shows an example of the two-level LSH structure where $L = 4$ and $C = 3$ (a, b, c, f are the hexadecimal notations for 10, 11, 12 and 15 respectively). The weights belong to {1,3,7,15}. Two WMCs are represented. WMC1 is assigned weights $H_1 =< f, 3, 1, 7 >$ at positions $P_1 =< 2, 4, 6, 9 >$. WMC2 is assigned weights $H_2 =< 3, 1, 7, f >$ at positions $P_2 =< 1, 2, 6, 9 >$. WMC1 and WMC2 collide in the bucket associated with the position subset <2,6,9>. The second level hash table has 24 blocks, each associated with a permutation of the weights. In the second level hash table, inside the bucket determined by the position subset <2,6,9>, WMC1 has the hash value $H_{1,17} =< f, 1, 7 >$ while WMC2 has the hash value $H_{2,13} =< 1, 7, f >$. Therefore, they are stored in different blocks.

4.4 Multi-Probing

Blocks in the second level hash table are non-overlapping. As a result, two WMCs with different weights cannot collide

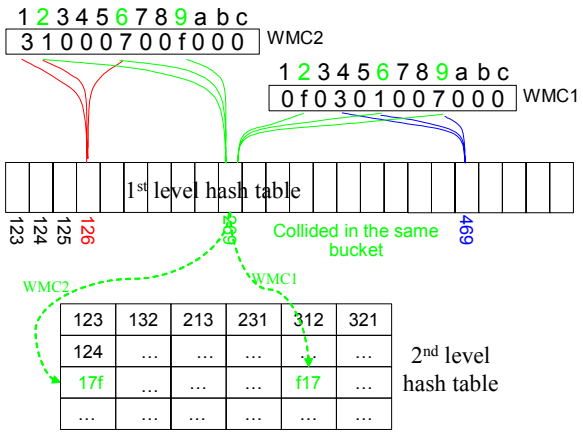

Figure 5: Two-level LSH structure.

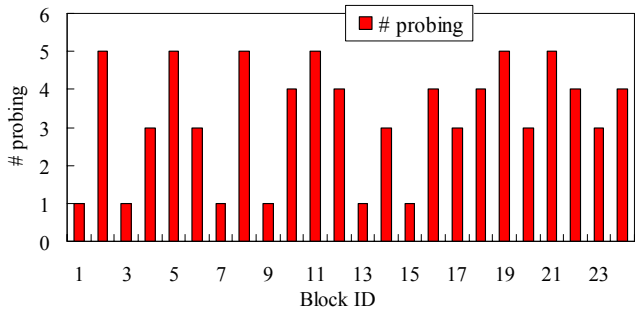

Figure 6: Number of probes per block.

in the same block even if their similarity is high. Improving the recall requires probing several blocks.

Each block is associated with a subset of weights and all WMCs inside the same block share these weights. The common weights can represent a block. Similarity between two blocks is calculated based on these weights, as shown below.

$$\varphi = \frac{\sum_i \min\{H_1(i), H_2(i)\}}{\sum_i \max\{H_1(i), H_2(i)\}} \quad (4)$$

If we use binary representations for the weights and concatenate these representations, then the above definition is equivalent to the Jaccard similarity coefficient [23].

With multi-probing, when a query matches a block, not only the features in that block, but also the features in the *relevant* neighboring blocks are selected as candidates. A neighbor of a block is relevant if its similarity with the block is above a predefined threshold φ_{th}. Since a block is defined by a permutation of the weights, the relevant neighbors of each block can be easily pre-determined.

Note that not all the blocks have the same number of relevant neighbors. Figure 6 shows the total number of blocks that should be probed for a query falling in one of the 24 blocks (with the two-level hash scheme in Figure 5). The horizontal axis gives the block ID; the block similarity threshold is $\varphi_{th} = 0.75$. Different numbers can be obtained for other thresholds.

4.5 Search Process

Given an audio query q as input, the retrieval of similar tracks (potentially versions of the query) from the database follows these steps:

(i) The audio query q is converted into a sequence, $q_1, q_2, \cdots$ of local summaries (WMC features).

(ii) For each WMC feature q_i, the positions and weights of its major bins, P_i and H_i, are determined. For each subset P_{ij} of P_i and H_{ij} of H_i, the corresponding bucket and then block are found. The WMC features in this block and in the neighboring blocks are retrieved. These form the rough candidates $\{r_{mn}\}$.

(iii) Now q_i is compared against $\{r_{mn}\}$ according to Eq.(1). Only WMCs with a correlation coefficient higher than the pre-determined threshold remain after this filtering. The remaining matching pairs are $\{< q_i, r_{mn}, \rho >\}$.

(iv) Matching pairs found with all the WMCs in the query are used to determine whether a candidate track is really relevant. With all matching pairs of the same reference track, a Hough transform is performed to check the linearity as in [18]. Here, q_i and r_{mn} are local summaries and associated with each is the offset of the corresponding CCP inside the audio track; the actual offset is used in the computation.

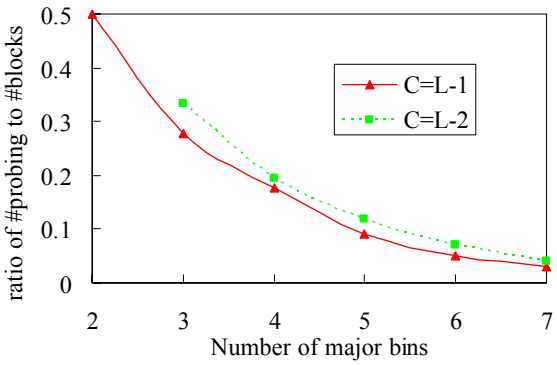

Figure 7: Ratio of #probing to #blocks in the second level hash table.

4.6 Speedup Analysis

Assume that, after local summarization, there are N WMCs in the database. Exhaustive search would require comparing a query WMC to each of the N WMCs.

Let us now evaluate the cost of processing the query with the two-level LSH scheme. In the first level hash table there are $\binom{V}{C}$ buckets and each feature appears in $\binom{L}{C}$ buckets. Therefore, on average, each hash bucket contains $N \cdot \binom{L}{C} / \binom{V}{C}$ WMCs. Each bucket is further divided into $\binom{L}{C} \cdot C!$ non-overlapping blocks at the second level of hashing. During retrieval W blocks are probed. It follows that the ratio between the number of similarity computations with the two-level hashing scheme and the number of similarity computations with exhaustive search is

$$\left[\binom{L}{C} \Big/ \binom{V}{C}\right] \times \left[W \Big/ \left[\binom{L}{C} C!\right]\right] \quad (5)$$

This is also the inverse of the expected speedup (acceleration). In the above equation V is a fixed value defined by the features employed, W depends on block similarity, while L and C ($C < L$) are in principle adjustable parameters.

The second factor in Eq.(5) is the probing ratio, i.e. the ratio of the number of probes to the total number of blocks

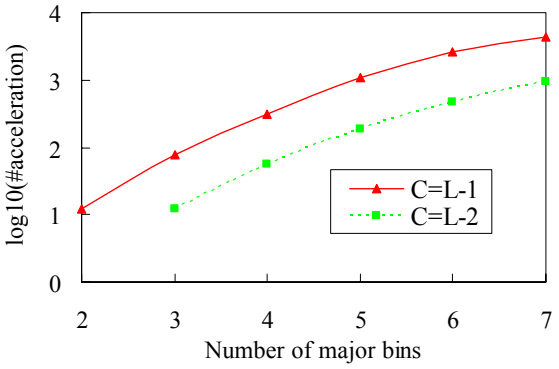

Figure 8: Speedup with two-level hash structure.

in the second level hash table. Figure 7 shows the average probing ratio for different numbers of major bins (L), with the block similarity threshold set to 0.75. Although there are more blocks as C increases (together with L), more blocks are similar and thus need to be probed. Therefore the probing ratio does not decrease much. The inverse of the probing ratio, $[(\frac{L}{C})\cdot C!/W]$, reflects the speedup obtained by two-level LSH with respect to single-level LSH.

Figure 8 shows the average acceleration following Eq.(5), with the average probing ratio given in Figure 7. According to Figure 8, when $L = 4$ and $C = 3$, the two-level hash structure can make retrieval 310.6 times faster. Although a larger L could lead to a higher acceleration, this is not used since the $L = 4$ major bins already contain the largest part of the Chroma energy according to Figure 3.

4.7 Hashing Refinement

In Figure 6, the number of probes for 6 blocks equals 1, indicating that these blocks have no relevant neighboring block. An investigation shows that these 6 blocks correspond to the subset of weights {1,3,7}, i.e., they are associated with a subset of weights with small values. A further study confirms that these blocks can be safely removed, to reduce both the storage and computation costs. Assume that two WMCs, WMC1 and WMC2, collide in some bucket/block associated with <1,3,7>. This can happen in two cases:

(i) The position of the 1^{st} major bin in WMC1 and WMC2 is different. Then the similarity between H_1 and H_2 is of only (1+3+7)/(1+3+7+15+15) = 11/41, which is much less than the total similarity threshold. This matching pair can be safely removed.

(ii) The position of the 1^{st} major bin in WMC1 and WMC2 is the same. Then WMC1 and WMC2 also collide in the same block associated with $<3,7,f>$ and can be retained for the last filtering stage in virtue of this last collision. Hence, the blocks associated with <1,3,7> are unnecessary.

In general, when the block similarity threshold is determined, some blocks can be omitted because they are associated with low-weight hash values.

5. EXPERIMENTAL RESULTS

We have conducted several experiments to evaluate the retrieval quality and the efficiency of the proposed method based on local summarization and multi-level LSH. We first introduce the experimental setup in section 5.1. The real audio datasets are described, then the ground truths, tasks and evaluation metrics are presented. Four experiments are performed to measure the effect of the spectral correlation threshold and the block similarity threshold, the robustness, and the effect of the query lengths, respectively.

Table 1: Datasets description.

Dataset	Covers79	ISMIR	RADIO	JPOP
#Tracks	1072	1458	1431	1314
Size	1.42GB	1.92GB	1.89GB	1.73GB

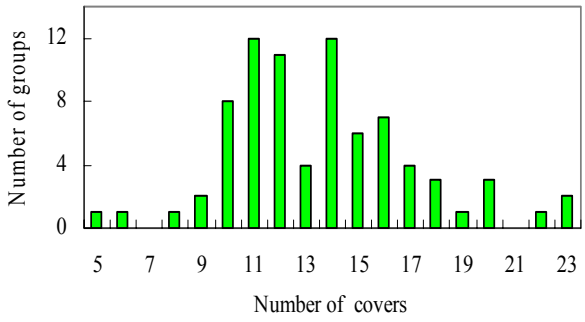

Figure 9: Distribution of cover tracks in Covers79.

5.1 Experiment Setup

Datasets. Our music collection (5275 tracks) consists of four non-overlapping audio track datasets, as summarized in Table 1. Covers79 is collected from `www.yyfc.com` and contains 79 popular Chinese songs, each present in several versions (same song interpreted by different persons). A song has on average 13.5 versions, resulting in a total of 1072 audio tracks. Figure 9 shows the distribution of these tracks as a function of the number of their covers (e.g., 12 songs have 11 covers each). The tracks in Cover79 were recorded by different users with simple computer microphones, so background noise is present in the recordings. In other words, the query is noisy, which makes the evaluation results meaningful for real applications.

The RADIO dataset was collected from `www.shoutcast.com`, while the ISMIR dataset was taken from `ismir2004.ismir.net/genre_contest/index.htm`. JPOP (Japanese popular songs) is from our personal collections. These three datasets are used as background audio files in our experiments. To further investigate the robustness of our algorithms, we also collected a real noise dataset (denoted by RNoise) that can be used as the queries' background noise. RNoise contains 396 noise tracks recorded in public places, for example in a bus on the highway, on the campus or in the subway.

Each track is 30s long in mono-channel wave format, the sampling rate is 22.05 KHz with 16 bits per sample. The audio data is normalized and then divided into overlapping frames. Each frame contains 1024 samples and the adjacent frames have 50% overlap. Each frame is weighted by a Hamming window and 1024 zeros are further appended. A 2048-point FFT is used to compute the STFT from which the instantaneous frequencies are extracted and Chroma is calculated, then the WMC features are obtained through local summarization.

Benchmark. The ground truth is set up according to human perception. We have listened to all the tracks and

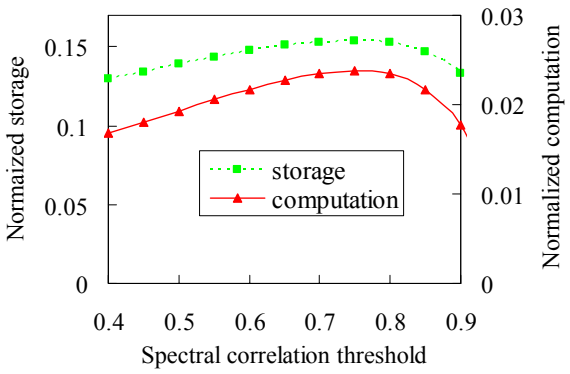

Figure 10: Storage and acceleration by local summarization.

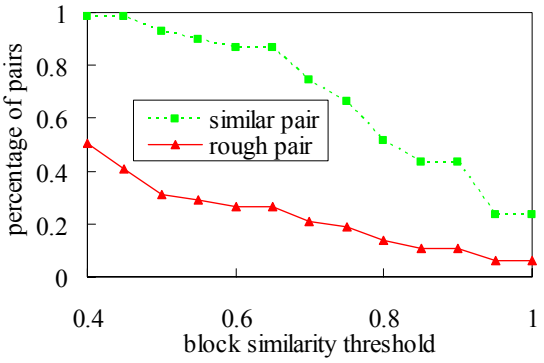

Figure 11: Percentage of matched pairs.

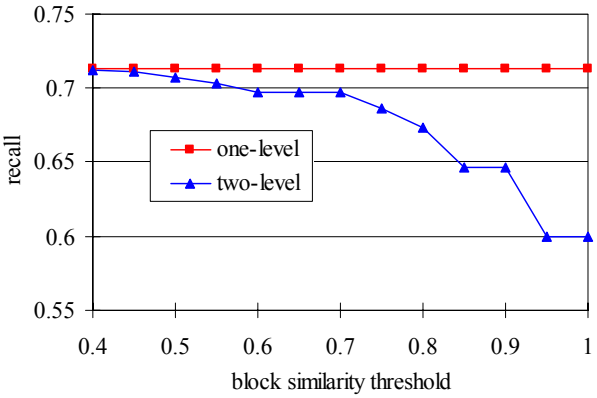

Figure 12: Recall for different block similarity thresholds.

manually labeled them, so the retrieval results of our algorithms should correspond to human perception in order to support practical applications. Covers79 is divided into groups according to the verses (the lyrics corresponding to the main theme) to judge whether the tracks belong to the same group or not. One group represents one song and the different versions of the song are members of this group. The 30 second segments in the dataset are extracted from the main theme of the songs.

Tasks. We consider the problem of cover songs detection or near duplicate detection of audio files, as in [1, 2, 3, 6, 14]. But we focus on the cases where the queries only match a part of the relevant references in the database. With a part of each track in Cover79 as the query, its cover versions are retrieved. The extra noise recorded in public places is also added to some queries to evaluate the robustness of the proposed retrieval scheme.

Evaluation metrics. We focus on recall as the main metric. Indeed, we want to see whether the adapted two-level LSH scheme we proposed is more selective (returns fewer irrelevant results) than baseline LSH but does not miss many more relevant results. Given a query q as musical audio input, S_q is the set of items that are relevant to this query in the database. As a response to the query, the system outputs the retrieved set K_q in a ranked list. In the following experiments $|K_q|$ equals $|S_q|$. Recall is defined as

$$\text{recall} = |S_q \cap K_q|/|S_q| \quad (6)$$

5.2 Effect of Spectral Correlation Threshold

In the local summarization stage, each audio track is divided into multiple CCPs, separated by isolated frames. Each CCP has a length of at least 2. Frames that are not similar to any adjacent frame are discarded (isolated frames). The number of CCP (WMC) depends on the spectral correlation threshold ρ_{th}. A large ρ_{th} produces many isolated frames and fewer CCP. On the other hand, with a small ρ_{th} two adjacent CCPs may be merged together. A proper ρ_{th} should maximize the number of CCPs.

A simulation result is shown in Figure 10, where the normalized storage δ is the ratio of the number of WMC (CCP) to the number of original frames. The value of δ reaches a maximum of 0.15 for ρ_{th}= 0.75. Since the equivalent length of each frame is 23 ms, the average duration of each CCP is of about (23 ms /δ) 150 ms.

Local summarization reduces the number of features not only in the database but also in the query, so the computation cost is reduced by a factor of $1/\delta^2$. The value of δ^2 is shown in Figure 10 as the normalized computation cost. For ρ_{th}= 0.75, δ^2 equals 2.37%, so retrieval is 42 times faster.

5.3 Effect of Block Similarity Threshold

Figures 11-12 show the results for different block similarity thresholds. Each query is on average 10 seconds long, or 1/3 of the length of its relevant songs. A "rough pair" is a pair that was found to match by the two-level LSH before the final filtering. A "similar pair" is a rough pair that remains after filtering. The numbers of rough pairs and similar pairs in two-level LSH are normalized, i.e. divided by the corresponding values in the baseline, single-level LSH. With a good design we expect that the percentage of similar pairs approaches 1 while the percentage of rough pairs approaches 0. As the block similarity threshold increases, both the percentage of rough pairs and similar pairs decreases. But for all cases the percentage of similar pairs is significantly higher than that of rough pairs, which confirms that the adapted two-level LSH is more selective than baseline LSH.

Because the percentage of similar pairs decreases when the block similarity threshold increases, the recall also decreases (Figure 12). The recall of the two-level LSH is very close to that of a single-level LSH when the block similarity threshold is no more than 0.7. Although from Figure 11 it may seem that the block similarity should be less than 0.65, the value

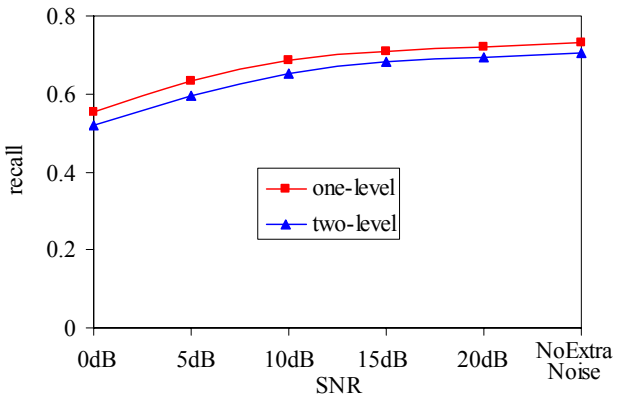

Figure 13: Recall under different SNR with non-white noise.

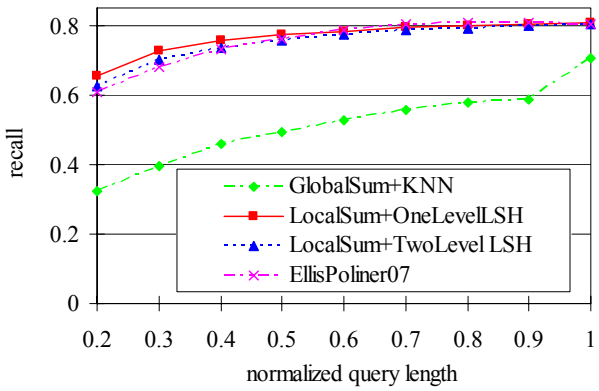

Figure 14: Recall for different query lengths.

of 0.7 is reached in Figure 12 since the sequence comparison using the Hough transform [24] also has a contribution.

5.4 Evaluation of Robustness

In real applications, the query musical signal may be mixed with environmental noise, which raises a challenge with respect to the robustness of our solution. In fact, several components of the proposed retrieval scheme contribute to an increased robustness. Specifically, WMC features are obtained by the weighted temporal integration of Chroma features during a CCP, which can reduce the sensitivity to impulse noise. Furthermore, by only retaining the 4 major bins, the quantization of WMC features can provide robustness to noise whose spectrum does not replace or exchange any of these bins.

To show the robustness of our scheme, 396 queries are randomly selected from Covers79 and the 396 noise segments from the RNoise set are respectively added to them, at several values of the SNR (signal to noise ratio), to simulate a real, noisy environment. Figure 13 shows that the recall at lower SNR is a little less than that at higher SNR. In the evaluation, non-white noise is used and SNR is calculated from the ratio of average signal energy to average noise energy. For a query with a low SNR, the spectrum of the non-white noise may be stronger than the desired signal and change the spectrum structure completely. Therefore the recall degraded slightly. However, for SNR above 10dB, the recalls of both one-level LSH and two-level LSH approach the steady value reached when no extra noise is added.

5.5 Effect of Query Length

Audio signals are not stationary, the statistic properties change between different segments. This is why we suggest the use of local summarization. Figure 14 shows recall values under different normalized query lengths. The length of a query is normalized by dividing it by the length of its relevant songs (variants) in the database. Here, "EllisPoliner07" relies on comparing sequences of beat-synchronous Chroma by exhaustive search [2]. "GlobalSum" is the scheme suggested in [6], which exploits global summarization.

It is obvious that for global summarization, the recall heavily depends on the query length, since the statistical properties change. Recall is relatively low when a query has a much shorter length than its variants in the database. But when local summarization is employed, the performance is less sensitive to query length. This is because the local summarization provides a more complete representation of the properties of a query and of its relevant tracks. Therefore, the proposed scheme still achieves high recall even when the query is very short. Figure 14 also shows that, with two-level LSH, recall is very close to the value obtained with single level LSH, especially when the query is relatively long.

Surprisingly, the two-level LSH scheme achieves a similar recall as EllisPoliner07 and even outperforms it a little when the query length is short. This is due to two factors: (i) EllisPoliner07 heavily depends on beat detection and uses cross-correlation instead of dynamic programming in calculating the similarity. Thus, its performance degrades if errors occur in beat detection. (ii) Our scheme is designed to be robust. Even when the noisy version significantly differs from the original track, they are still likely to share the same major bins over many frames, which ensures a relatively high recall.

6. CONCLUSION AND FUTURE WORK

The presence of large collections of multi-variant musical audio tracks on user-generated content websites and the interest such collections have for many users motivate work on the detection of the audio variants, either for directly answering user queries or for structuring the content of the collections. Since the textual annotations are frequently inappropriate or even missing, finding the variants of a song must rely on the audio content itself. However, this is not an easy task. The comparison of feature sequences get accurate retrieval but does not scale well because matching long sequences is expensive. Alternatively, very compact descriptions of entire audio sequences support good scalability but retrieval accuracy is limited. Achieving a good balance between accuracy and efficiency is an important problem in detecting and grouping multi-variant audio tracks.

To obtain both accurate and efficient retrieval, in this paper we proposed a new method combining local summarization and multi-level locality-sensitive hashing. Based on spectral similarity, we suggest dividing each audio track into multiple continuously correlated periods of variable length. By removing a significant part of the redundant information, this provides support for more compact yet accurate descriptions. Weighted mean Chroma features are computed

for each of these periods. Then, by exploiting the characteristics of the content description, we define an adapted two-level locality-sensitive hashing scheme for efficiently delineating a narrow relevant search region. At the first level a "coarse" hashing is performed in order to restrict search to items having a non-negligible similarity to the query. To find the items that are highly similar to the query, a subsequent "refined" hashing is used.

Our analysis shows that a significant speedup can be expected. We believe that this multi-level hashing scheme can be further improved by an adapted representation of bucket content and by directly taking the longer-range temporal information into account.

We presented evaluation results and compared the accuracy and efficiency of our method in retrieving multi-variant audio tracks. We have shown the practical application of the proposed algorithms via experiments on a multi-variant music dataset, with a ground truth based on human perception. The results of cover song detection demonstrate that (i) local summarization far outperforms global summarization; (ii) the adapted two-level LSH scheme significantly improves query selectivity compared to conventional LSH. Together, these proposals achieve a much better tradeoff between retrieval accuracy and efficiency.

The method put forward here can be directly employed for answering queries by example, but can also serve for grouping together the different variants of the audio tracks in the database. We believe this methodology can be extended to other types of problems in music information retrieval, by using adequate features and similarity measures.

7. ACKNOWLEDGMENTS

This research was supported by a grant from the Department of Defense through the KIMCOE Center of Excellence. We thank Heng Lu for helping collect the RNoise dataset.

8. REFERENCES

[1] J. Serra, E. Gomez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Trans. on Audio, Speech and Language Processing*, 16(6):1138–1152, Aug 2008.

[2] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proc. ICASSP'07*, volume 4, pages 1429–1432, 2007.

[3] M. Casey and M. Slaney. Song intersection by approximate nearest neighbor search. In *Proc. ISMIR'06*, pages 144–149, 2006.

[4] Y. Yu, K. Joe, and J. S. Downie. Efficient query-by-content audio retrieval by locality sensitive hashing and partial sequence comparison. *IEICE Trans. on Information and Systems*, E91-D(6):1730–1739, Jun 2008.

[5] M. Lesaffre and M. Leman. Using fuzzy to handle semantic descriptions of music in a content-based retrieval system. In *Proc. LSAS'06*, pages 43–54, 2006.

[6] Y. Yu, J. S. Downie, F. Moerchen, L. Chen, and K. Joe. Using exact locality sensitive mapping to group and detect audio-based cover songs. In *Proc. IEEE ISM'08*, pages 302–309, 2008.

[7] B. Cui, J. Shen, G. Cong, H. Shen, and C. Yu. Exploring composite acoustic features for efficient music similarity query. In *Proc. ACM MM'06*, pages 634–642, 2006.

[8] F. Moerchen, I. Mierswa, and A. Ultsch. Understandable models of music collection based on exhaustive feature generation with temporal statistics. In *Proc. ACM KDD'06*, pages 882–891, 2006.

[9] W. H. Tsai, H. M. Yu, and H. M. Wang. A query-by-example technique for retrieving cover versions of popular songs with similar melodies. In *Proc. ISMIR'05*, pages 183–190, 2005.

[10] R. Miotto and N. Orio. A methodology for the segmentation and identification of music works. In *Proc. ISMIR'07*, pages 239–244, 2007.

[11] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multiprobe lsh: efficient indexing for high dimensional similarity search. In *Proc. VLDB'07*, pages 950–961, 2007.

[12] N. C. Maddage, H. Li, and M. S. Kankanhalli. Music structure based vector space retrieval. In *Proc. ACM SIGIR'06*, pages 67–74, 2006.

[13] T. Pohle, M. Schedl, P. Knees, and G. Widmer. Automatically adapting the structure of audio similarity spaces. In *Proc. 1st Workshop on Learning the Semantics of Audio Signals (LSAS)*, pages 66–75, 2006.

[14] J. P. Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proc. ISMIR'07*, pages 239–244, 2007.

[15] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and Y. Manolopoulos. Audio indexing for efficient music information retrieval. In *Proc. MMM'05*, pages 22–29, 2005.

[16] J. Reiss, J. J. Aucouturier, and M. Sandler. Efficient multi dimensional searching routines for music information retrieval. In *Proc. ISMIR'01*, pages 15–20, 2001.

[17] N. Bertin and A. Cheveigne. Scalable metadata and quick retrieval of audio signals. In *Proc. ISMIR'05*, pages 238–244, 2005.

[18] C. Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *Proc. ACM MM'02*, pages 584–591, 2002.

[19] Y. Yu, C. Watanabe, and K. Joe. Towards a fast and efficient match algorithm for content-based music retrieval on acoustic data. In *Proc. ISMIR'05*, pages 696–701, 2005.

[20] M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Trans. on Multimedia*, 7(1):96–104, Feb 2005.

[21] P. Indyk and N. Thaper. Fast color image retrieval via embeddings. In *Proc. Workshop on Statistical and Computational Theories of Vision*, 2003.

[22] S. Hu. Efficient video retrieval by locality sensitive hashing. In *Proc. ICASSP'05*, pages 449–452, 2005.

[23] http://en.wikipedia.org/wiki/jaccard_index.

[24] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, 1995.