Combining Multi-Probe Histogram and Order-Statistics Based LSH for Scalable Audio Content Retrieval

Yi Yu*
Dept. of Computer Science
New Jersey Inst. of Tech.
Newark, NJ, 07102, USA
yuyi@njit.edu

Michel Crucianu
CEDRIC - CNAM
292 rue St Martin
75141 Paris cedex 03, France
michel.crucianu@cnam.fr

Vincent Oria
Dept. of Computer Science
New Jersey Inst. of Tech.
Newark, NJ, 07102, USA
oria@njit.edu

Ernesto Damiani
Dip. di Tecnologie
dell'Informazione, Università
degli Studi di Milano, Italy
ernesto.damiani@unimi.it

ABSTRACT

In order to improve the reliability and the scalability of content-based retrieval of variant audio tracks from large music databases, we suggest a new multi-stage LSH scheme that consists in (i) extracting compact but accurate representations from audio tracks by exploiting the LSH idea to summarize audio tracks, and (ii) adequately organizing the resulting representations in LSH tables, retaining almost the same accuracy as an exact kNN retrieval. In the first stage, we use major bins of successive chroma features to calculate a multi-probe histogram (MPH) that is concise but retains the information about local temporal correlations. In the second stage, based on the order statistics (OS) of the MPH, we propose a new LSH scheme, OS-LSH, to organize and probe the histograms. The representation and organization of the audio tracks are storage efficient and support robust and scalable retrieval. Extensive experiments over a large dataset with 30,000 real audio tracks confirm the effectiveness and efficiency of the proposed scheme.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*modeling*

General Terms

Algorithms, Performance, Experimentation

*This work was performed while the first author was visiting University of Milan (Università degli Studi di Milano).

MM'10, October 25–29, 2010, Firenze, Italy.

Keywords

Variant audio search, multi-probe histogram, locality sensitive hashing, audio computing, music-IR, order statistics

1. INTRODUCTION

In the era of social networks, people can upload their favorite popular music or lip-synch music performance with the original singer's background to music social websites and share them with their friends. This also allows spammers to post several times the same irrelevant audio track with the semantic tags of a popular song, aiming at boosting its ranking. Besides, the titles and semantic tags are often noisy since users might type ambiguous or incomplete information. These negative effects of users activities make it difficult to classify and retrieve songs by semantic tags over social networks, especially for large collections having very diverse contents and contributors.

Instead, *content-based* music variant detection and retrieval is getting more and more important and it is a natural way of searching. Audio content analysis allows to filter out audio spam when the tags are missing, ambiguous or even wrong. Unfortunately, it suffers from the expensive feature sequence matching that hinders the scalability of music content retrieval from large databases (e.g. last.fm and YouTube music channel have millions of music tracks). To address this issue, recent studies have focused on indexing acoustic features and organizing large music recording databases for efficient access [1, 2, 3, 4, 5]. Locality Sensitive Hashing (LSH) [6] has been widely employed in the recent years to support scalable multimedia retrieval and mining. An actual LSH design for musical content depends on the representations of musical audio features. A brief survey of scalable music content retrieval can be found in [7].

In this paper, we use the LSH concept in two different stages for reliable and scalable content-based music retrieval. In the first stage, an entire audio sequence is summarized by calculating a *multi-probe histogram* (MPH). In the second stage, according to the order statistics (OS) of MPH, an adapted LSH scheme, OS-LSH, is put forward to map the MPHs to hash keys. Specifically, the bins having the highest energy (major bins) in the chroma features are detected. Major bins from adjacent frames are then concatenated as

phrases, from which a MPH is calculated as the representation of an entire audio track. The combinations of major bins from adjacent frames represent well the significant information regarding both spectral energy and local temporal correlations of audio signals. MPHs are further stored in LSH tables according to the hash values calculated from the statistic properties of MPHs. We consider various versions of the same song as "music copy", for example lip-synch, duplicate or near-duplicate music recordings, variant music tracks recorded for the same song by different people with similar musical settings. To evaluate the effectiveness and the scalability of the proposed algorithms, we performed experiments on a dataset consisting of 30,000 audio tracks. We compared our solution to several existing methods. The extensive experiments confirm that our approach more effectively solves the scalability issue.

We start by introducing the research background and review related work in section 2. Robust audio features are discussed in section 3. Then, we entirely describe the new retrieval approach and discuss parameter values in section 4. The experimental setup is shown and the evaluation results are reported in section 5. Section 6 concludes with a summary of this work.

2. BACKGROUND AND RELATED WORK

This work addresses the issue of scalable query-by-audio Music Information Retrieval (MIR) following a two-stage approach where each stage relies on LSH. In the following we introduce the research background for query-by-audio MIR, LSH variants and acoustic matching via LSH. Then we summarize the related work with a comparative perspective.

Query-by-audio. MIR has gradually developed into an interdisciplinary research field that is related to audio signal processing [8, 9], audio content indexing [1, 2, 3, 4, 7, 10], sequence matching [8], pattern recognition [11], music retrieval evaluation [12], etc. Applications of query-by-audio MIR include near duplicate audio detection [13], audio-based music plagiarism analysis [14] and query-by-example/humming/singing [1, 10]. Although different issues such as search intention, music genre and mood, personal interests and culture background, have to be considered during the search formulation, the retrieval process is generally simple: take audio content as the query, perform similarity search and finally return the results in a ranked list. Audio sequences contain rich information described by high-dimensional features [15, 16]. A scalable music retrieval and mining solution necessitates a reliable and concise representation of relevant acoustic information, and an efficient organization of these representations.

LSH principle and variants. A hash function associates a *hash key* to every data item. The function is said to be *locality sensitive* if items close to each other have a high collision probability while those far from each other have a low collision probability. To retrieve by similarity, one has to compute (off-line) the hash keys for all the items in the database; the items having the same key are stored together in the same *bucket*. Then, when a query item arrives, its hash key is also computed and all the items in the bucket corresponding to this key are returned. LSH is widely used for index-based data organization and supports a scalable approximate nearest neighbor retrieval over a large database [6]. Some LSH variants have been proposed to improve retrieval quality by performing multi-probe [17, 18] or spectral-hashing [19]. LSH-based proposals were also put forward for actual scalable multimedia content retrieval and mining (audio [1], image [20, 21], video [22]).

Music sequence representation for scalable matching. There is significant research interest in finding audio signal representations and indexing solutions capable of supporting scalable musical retrieval and mining. The difficulty comes from the fact that the audio content is usually described by a sequence of high-dimensional features. Three approaches for representing audio signals exist: global summarization [2], use of a sequence of features [1] and local summarization [3, 7]. In [2], frequently employed audio features (MFCC, Chroma, Mel-magnitudes and Pitch) are combined together using trained weights, and a long audio sequence is summarized into a single, compact and semantically rich, Feature Union (FU). SoftLSH and Exact Locality Sensitive Mapping (ELSM) are then suggested to accurately locate the search range. In his seminal work on audio content searching [1], Yang employed random subsets of spectral features (STFT) to calculate the hash values for the multiple LSH hash instances in parallel. The original acoustic feature sequences are converted into indexable items and then stored in hash tables. To address the bucket conflict issues, a Hough transform is implemented on these matching pairs to detect the similarity between the query and each reference. In [3], log-frequency cepstral coefficients (LFCC) and chromagram features are extracted to represent the audio sequences. Audio shingles are created by concatenating consecutive frames. Then, LSH is adopted for approximate nearest neighbor searches. Weighted Mean Chroma (WMC) is proposed in [7] to generate local temporal summaries with low information loss. The WMCs are stored in a two-level LSH table. At the first level, a "coarse" hashing is performed to restrict search to items having a non-negligible similarity to the query. At the second level, a "refined" hashing is used to find items that are highly similar to the query.

In general, it is difficult to retain the relevant information in an audio with a global summarization. On the other hand, a comparison of feature sequences [1] allows exploiting the relevant temporal information but is computationally much more expensive as it has to combine frame matching and sequence matching. While the local summarization of acoustic features in [3, 7] significantly reduces the number of frames to be matched, the *a posteriori* comparison of sequences is still time consuming. Although the state of the art shows that LSH could help speed up content-based music retrieval, scalability remains a challenge. To that end, more relevant information on the audio tracks, including their temporal structures should be extracted and concisely represented to ensure reliable audio similarity.

The proposal presented below distinguishes itself from existing work by the fact that it computes compact global summaries that retain both spectral and temporal information from the audio tracks, and compares these summaries according to an adapted LSH scheme. More specifically, this work exploits LSH in two stages. In the first stage, correlations between major bins of chroma features from adjacent frames are cumulated in a multi-probe histogram (MPH) for a concise representation of the entire audio track. The robustness of this representation is improved by probing multiple chroma bins. In the second stage, the order statistics of MPH are analyzed and, on this basis, MPHs are organized and retrieved via an adapted LSH scheme.

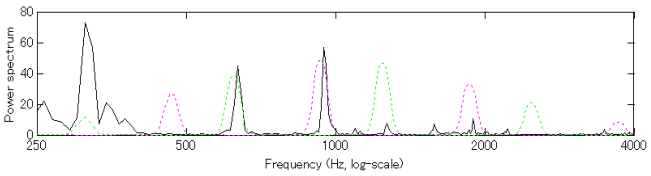

Figure 1: Music harmonic structure.

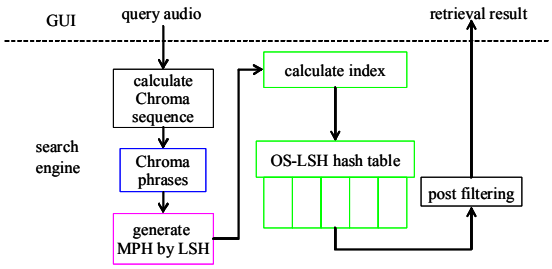

Figure 2: Retrieval with MPH and OS-LSH.

3. AUDIO REPRESENTATION FOR LSH

Songs can be represented by a sequence of acoustic features. These features are high-dimensional and their temporal arrangements are needed for accurate retrievals. But the direct comparison of the acoustic feature sequences is computationally expensive and the similarity-based retrieval from massive datasets is very challenging. Finding the appropriate feature representations is the key to the solution. Our solution is based on the chroma feature and LSH. In this section we review the chroma feature and discuss about how to exploit it in robust LSH schemes.

3.1 Spectral Information

Conventionally, pitch is associated with music notes and is often used to represent audio signals. The entire frequency band of the signal is divided equally in the octave scale into 88 sub-bands so that the central frequency of each sub-band is $2^{1/12}$ times that of the previous sub-band. Each sub-band corresponds to a pitch. Unlike other audio signals, music signals have obvious harmonic structures in their spectrum. When many harmonics are present, it is not easy to determine the actual pitch, even when sub-harmonic summation [23] is employed. Moreover, multiple simultaneous notes are not well represented by a single pitch.

Fig. 1 shows the spectral structure of one frame. In this figure, the spectral profile can be divided into two frequency families. Each frequency family represents a series of harmonics. These harmonics, although different in frequency, are usually perceptually similar to users. Instead of distinguishing the harmonics, the chroma feature calculates the total energy of the family of harmonics. The whole spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave that correspond to 12 pitch classes.

Chroma is a time-honored technique for representing musical audio content in tasks like variant detection and song identification [8, 9]. Because they identify "semantic" spectral components and distinguish them by music octave [23], chroma features are shown to be robust to the differences between variant audio recordings of the same song, concerning timbre, dynamics, tempo, etc, [8]. However, if the sequence of chroma features is directly used, the retrieval system does not scale with the audio database size. Since chroma represents the spectral structure and adjacent chroma features have strong correlations, a chroma sequence shows significant redundancy. We aim at removing chroma's unnecessary redundancy while preserving its capability of discriminating music variants.

3.2 Temporal Information

The melodic information in music is provided by the temporal evolution of the spectrum. The audio signal can be divided into successive frames. For each frame, various spectral features (Chroma, Pitch, MFCC, etc.) can be calculated. Dynamic programming can be used for comparing sequences while taking potential tempo variations into account. But this solution is computationally expensive and does not scale well to long sequences. To find a LSH-based solution that does not require *a posteriori* sequence matching, one has to answer the following question: how to convert a sequence of acoustic features into efficiently indexable items that retain the relevant similarities regarding both spectral and temporal characteristics.

A straightforward strategy is to tie adjacent frames together and retain *local* temporal correlations. We expect this additional temporal information, stored in a compact format, to significantly increase the discriminative power as compared to the use of spectral information alone. To implement this idea, the local temporal information is extracted from correlations between spectral features of adjacent frames and stored in a global histogram. A multi-probing mechanism is introduced to improve the robustness of the representation. Following the order statistics of these histograms, a new LSH scheme is designed to achieve good time efficiency for audio content retrieval.

4. MULTI-STAGE LSH ALGORITHMS

This section describes the solution we are proposing for performing an efficient and effective content-based retrieval of variant audio tracks, by exploiting LSH in two successive stages. Fig. 2 provides an overview of the system that includes the user interface and the search engine. The search engine extracts a sequence of chroma from the audio query as the initial feature. This chroma sequence is further summarized to a concise multi-probe histogram (MPH) that is mapped into keys in the LSH tables. Candidate histograms in the buckets associated with these keys are compared against the query in a post-filtering step and only the similar items are returned to the user in a ranked list. In the following subsections, we elaborate on how to generate MPH from the set of adjacent frames in an audio sequence and how to arrange them in the hash table via the OS-LSH scheme to realize a scalable music audio content retrieval system. We also provide an analysis showing how to select the values of the parameters of this method. To this end, some key parameters of the proposed method are tuned by experiments using TuneSet, a dataset which contains 1,200 tracks having diverse content with different types (collected from the music channel of Youtube), 200 being used as queries and 1,000 as background.

4.1 Multi-Probe Chroma Histogram

The first stage of our approach addresses the problem of

summarizing a sequence of acoustic features into a compact music representation that retains most relevant similarities between acoustic sequences. We aim to avoid the computationally expensive matching of long sequences of spectral features, needed for taking complete melodic information into account. Instead, we focus on the *statistical* representation of acoustic *short-range* sequential correlations, which should provide a more adequate balance between scalability, robustness and discrimination ability. We apply the principle of LSH to build a *multi-probe histogram* from a sequence of chroma features.

4.1.1 Quantization of Chroma Features

As discussed in section 3.1, chroma is an appropriate representation for spectral properties of audio signals in variant audio track retrieval. The audio sequence is divided into overlapping frames. For each frame, a 12-bin chroma feature is computed. Each bin represents the energy of a semi-tone (or frequency family). It was found in [7] that, on average, the 4 major bins (having the highest energy) account for more than 80% of the energy in the chroma features. The other bins have a comparatively low contribution to the similarity between the chroma features and can be neglected. Moreover, a coarse quantization of the energy in the first four bins does not have a significant impact on the similarity either. Similarity can then be computed by translating these coarse quantizations into relative weights. We also apply these special characteristics to calculate a compact representation from the chroma sequence.

4.1.2 Chroma Phrases

To improve the discrimination power among audio tracks while keeping their representations compact, we only exploit short-range temporal information for an audio sequence. More specifically, we consider short sequences of J adjacent frames with their chroma features. When each chroma feature is represented by its major bins, adjacent frames can be described as a sequence of major bins, refereed to as a *phrase*. Such a phrase captures the local temporal characteristics of the audio signal. As an example, if each chroma feature is represented on K=4 bins and J=2, we obtain K^J=16 phrases for each sequence of consecutive J frames.

4.1.3 Multi-Probe Histogram

It is necessary to further compute some statistics from these phrases to generate a fixed-size, low-dimensional and concise summary of the audio sequence. For this purpose, the aforementioned phrases are mapped to the bins of a histogram that counts the frequency of each phrase.

Perceptually similar audio tracks may have slightly different spectral structures. Therefore, it is not necessary that they have the same set of major chroma bins. But they should have most major chroma bins in common. To consider such potential differences, a multi-probing approach is adopted. From adjacent frames, multiple phrases, corresponding to different combinations of these major chroma bins, are calculated. Probing the less important major chroma bins makes the MPH more robust and enables significant improvement of the recall.

Major chroma bins do not have the same energy. Since it is the relative strength of bins that determines the perceptual similarity, we use weights, rather than the original energy, to represent the major bins. As each phrase consists of chroma

Procedure calculateMultiProbeHistogram(s_i)
1. Divide the audio track s_i into frames $f_{i,j}$, j=1,2,…
2. For each frame $f_{i,j}$ of s_i
3. Calculate the chroma feature $c_{i,j}$ from $f_{i,j}$
4. Sort the 12-D $c_{i,j}$ by decreasing energy, the original positions of top K major bins are $m_{i,j,k}$, k=1,2,…,K
5. Assign weights $w_{i,j,k}$, k=1,2,…,K to these major bins
6. Initialize an all zero multi-probe histogram MPH_i
7. For j=1,…, $|c_i|$-J+1
8. $[k_1, k_2, …, k_J]$ = [1, 1, ..., 1]
9. For l=1,…,K^J
10. For r=1,...,J
11. Choose k_r^{th} major bin from $c_{i,j+r-1}$ as $m_{i,j+r-1,kr}$
12. Assign weight $w_{i,j+r-1,kr}$ to $m_{i,j+r-1,kr}$
13. Organize $[m_{i,j,k1}, …, m_{i,j+J-1,kJ}]$ into a phrase $P_{i,j,l}$
14. Organize $[w_{i,j,k1}, …, w_{i,j+J-1,kJ}]$ into a tuple $W_{i,j,l}$
15. Calculate the MPH bin as p=$[m_{i,j,k1}, m_{i,j+1,k2},…]_{12}$
16. Calculate the sum of weights as w=$w_{i,j,k1}$+$w_{i,j+1,k2}$+…
17. $MPH_i(p) = MPH_i(p) + w$
18. Set $[k_1, k_2, …, k_J]$ to the next combination of major bins
19. Normalize MPH_i

Figure 3: Computation of multi-probe histograms.

bins from multiple successive frames, each bin having its own weight, the sum of their weights is used as the weight of the phrase.

The detailed algorithm for computing the MPH of the audio track s_i is shown in Fig. 3. J is the number of consecutive frames that are concatenated to form phrases and K is the number of major chroma bins for each frame. At first, the audio track s_i is divided into multiple frames $f_{i,j}$ (line 1). For each frame $f_{i,j}$ of s_i, its 12-dimension chroma feature $c_{i,j}$ is calculated (lines 2-3). Then $c_{i,j}$ is sorted by decreasing energy of chroma bins, and the original positions of its top K bins are denoted as $m_{i,j,k}$, $k = 1, 2, \cdots, K$ (line 4). The weights assigned to these major bins are respectively $w_{i,j,k}$, $k = 1, 2, \cdots, K$ (line 5). Then, the MPH of s_i is calculated. For each position j in the sequence s_i, J adjacent frames are considered and all the K^J possible phrases of length J are built. Each phrase is processed as follows: the pointers $[k_1, k_2, \cdots, k_J]$, identifying the possible combinations of major bins from J frames, are initialized as $[1, 1, \cdots, 1]$ (line 8). By taking the bin identified by k_r from $c_{i,j+r-1}$ (lines 10-11) and concatenating these J bins in order, the l^{th} phrase $P_{i,j,l}$, is formed (line 13), together with its J-tuple weight $W_{i,j,l}$ (line 14). A hash value p is calculated from $P_{i,j,l}$ as the 12-base integer (line 15) according to Equation (1), where 12 is the dimension of chroma features, determined by music theory. This hash value p identifies one of the 12^J MPH bins. The weight w of the phrase $P_{i,j,l}$ is calculated as the sum of the weights inside $W_{i,j,l}$ (line 16), and is added to the MPH bin identified by p (line 17). Then $[k_1, k_2, \cdots, k_J]$ is adjusted to identify the next combination of major bins. In this process, each phrase is used only once. Finally the MPH_i is normalized so that the sum of all its bins equals 1 (line 19).

$$p=[m_{i,j,k1}, \ldots, m_{i,j+J-1,kJ}]_{12}=\sum_{r=1}^{J} m_{i,j+r-1,kr} \cdot 12^{J-r} \quad (1)$$

Fig. 4 shows an example where the number of adjacent frames is J=2, the number of major bins for each frame is

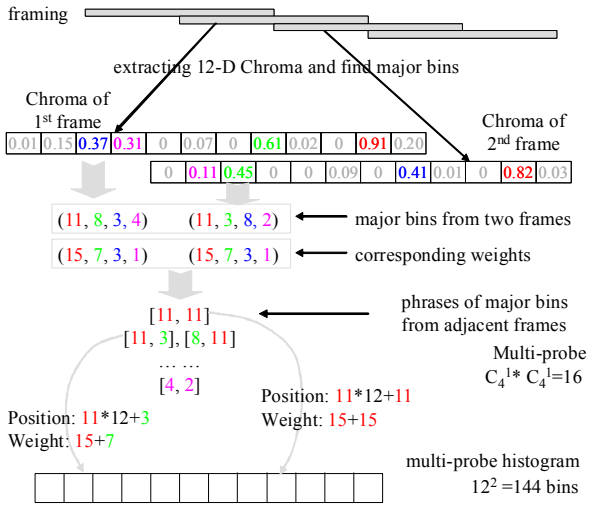

Figure 4: Simple example of multi-probe histogram.

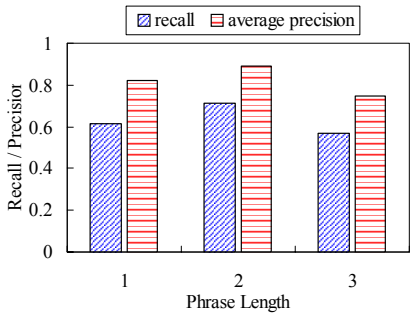

Figure 5: Impact of phrase length.

K=4, and the weights associated to the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} bin are respectively 15, 7, 3 and 1. The chroma features of two of the frames are calculated. The positions of major bins in the 1^{st} frame are (11, 8, 3, 4) and in the 2^{nd} frame are (11, 3, 8, 2). A phrase is constructed with one major bin from each frame. Altogether there are 2^4=16 combinations. The phrase [11,11] has the largest weight, 15+15, and this weight is added to the histogram bin corresponding to the 12-base integer $[11,11]_{12}$. Another phrase [11,3], with the total weight 15+7, is added to the histogram bin corresponding to the 12-base integer $[11,3]_{12}$.

4.1.4 *Phrase Length*

The phrase length J plays an important role in the MPH. Each J-phrase determines a hash value. Without further optimization, the dimension of a MPH is 12^J. A large J makes the summary discriminative and results in a high dimensional MPH. But large J values can also make the MPH too sensitive and affect the recall due to potential longer-range temporal differences between similar tracks. Using the TuneSet data, we evaluated the recall and average precision for different phrase lengths J, for exact kNN retrieval. According to Fig. 5, when J=2, both the recall and the average precision [24] reach their peaks; so J=2 is used hereafter.

4.1.5 *Number of Bins in Chroma Features*

The next important factor in MPH is the multi-probing. It depends on both the number of major chroma bins and their weights. According to [7], the weights $2^{K-k+1}-1$, $k=1,2,\cdots,K$, are used to differentiate the major bins with different energy. Fig. 6 shows the recall and average precision for different values of K on the TuneSet dataset. The recall and average precision increase with K because multi-probe makes the MPH more robust. When K increases above 4, the recall and average precision slightly decrease, so K=4 is used hereafter.

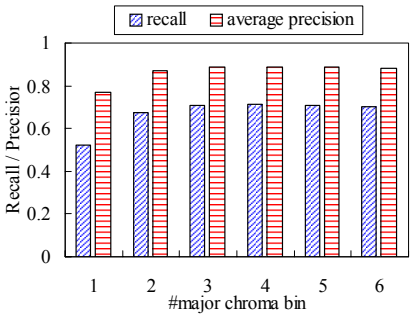

Figure 6: Impact of number of major chroma bins.

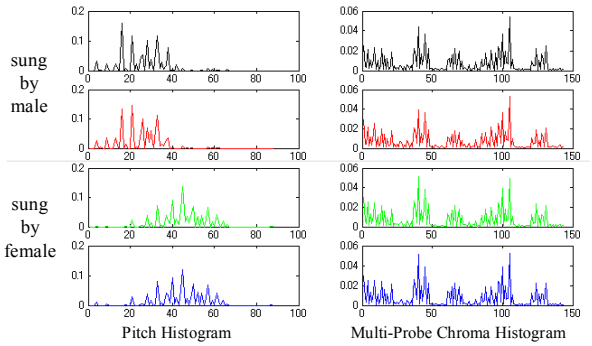

Figure 7: Pitch histogram vs. chroma MPH.

It is interesting to compare Fig. 5 with Fig. 6. The recall is only 0.52 when K=1 in Fig. 6, which is less than the recall obtained for J=1 in Fig. 5. This confirms that, as more frames are concatenated, the MPH becomes more discriminative and multi-probing is necessary to maintain a high recall.

4.1.6 *Chroma vs. Pitch Histograms*

Pitch histograms also reflect spectral statistics of audio signals. It is necessary to compare pitch histogram with MPH. Fig. 7 shows such a comparison. The left side and right side are respectively the pitch histogram and the MPH of 4 variants of the same song. These versions are sung by different people, two men and two women. It is well known that pitch is usually higher for a female than for a male. In this figure, the pitch for the females is roughly the double of that for the males. Pitch histograms separate the versions into 2 groups. With chroma, all histograms are very similar since the harmonics belong to the same frequency family.

4.2 Histogram Refinement

There are two types of chroma phrases. A type I phrase captures the stable spectral information in a music note because the positions of the major chroma bins are the same. In a type II phrase, at least one of the J chroma major bins is different from the others. A type II phrase captures a potential transition between consecutive music notes.

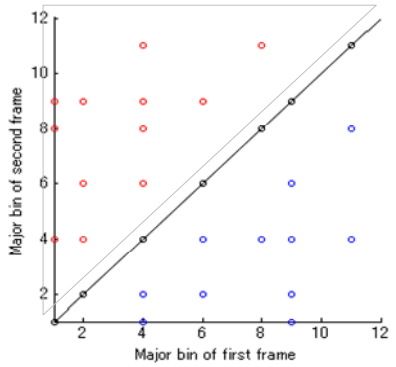

Figure 8: Distribution of pairs of chroma bins of consecutive frames.

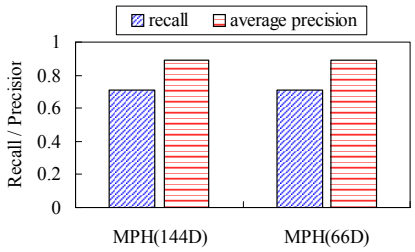

Figure 9: Refined vs. original chroma MPH.

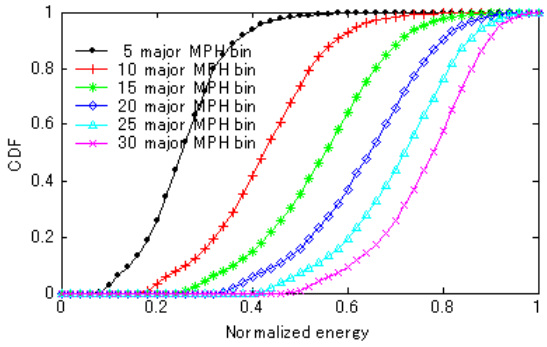

Figure 10: Cumulative density of major MPH bins.

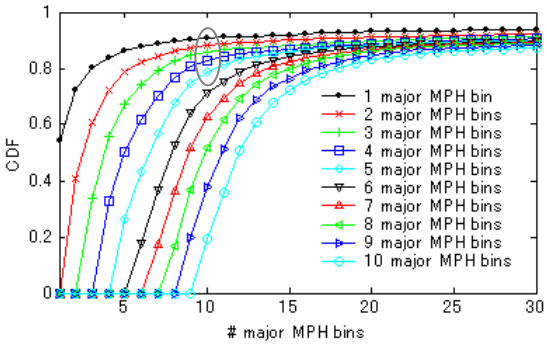

Figure 11: Cumulative density of major MPH bins.

Each MPH bin corresponds to a J-phrase of chroma major bins. According to Fig. 7 (right), each MPH has local peaks that represent the main profile. When J equals 2, the distribution of pairs of consecutive chroma bins exhibits some nice properties. Fig. 8 shows this distribution for the 40 major bins of a MPH: the axes represent the 12 dimensions of the chroma features of one frame vs. the next frame. The distribution is nearly symmetric with respect to the first diagonal. MPHs calculated from other audio tracks shows similar regularities. This is due to the fact that adjacent frames have some major chroma bins in common. For example, when the j^{th} and $j+1^{th}$ frames of the i^{th} audio track share the k_1^{th} and k_2^{th} major bins, the phrases $[m_{i,j,k_1}, m_{i,j+1,k_2}]$ and $[m_{i,j,k_2}, m_{i,j+1,k_1}]$ are both counted, with the same weight. Thus, a MPH is redundant and only half of its bins are necessary. In addition, a type I phrase records the stable (redundant) spectral information, so only type II phrases are necessary to discriminate the audio tracks. As a result, from the original 144-bin MPH, only the 66 bins in the triangle shown in Fig. 8 are necessary. This dimension is even lower than that of a pitch histogram, i.e. 88.

Fig. 9 shows the result of a comparison between the refined MPH(66D) and the original MPH(144D). MPH(66D) has similar average precision and a slightly higher recall than MPH(144D) because the less discriminating spectral information reflected by type I phrases is removed.

4.3 Design of Order Statistics-Based LSH

To improve scalability over exhaustive kNN retrieval, in the second stage the MPHs are organized according to an LSH scheme based on the order statistics of MPH.

4.3.1 Analysis of MPH

Each audio track has some distinguishing peaks in its MPH (see Fig. 7, right). These major MPH bins have significant influence on the cosine similarity used to evaluate the similarity between MPHs. We study how much of a MPH is represented by the major bins by sorting the MPH bins in decreasing order of their values. Fig. 10 shows the cumulative density function (CDF) of the sum of n major MPH bins, for $n = 5, \ldots, 30$. The CDF only becomes greater than 80% when $n = 30$ major bins (out of 66) are cumulated, which shows that no bin truly dominates.

Let us study the correlations between the major MPH bins of similar audio tracks, i.e. investigate the probability with which the top k major MPH bins of an audio track appear in the top n major MPH bins of its similar audio tracks. Fig. 11 shows the CDF curves for different values of k, the horizontal axis corresponding to n. The probability increases as k gets closer to n, showing that the major MPH bins of similar audio tracks do have strong correlations. When n=10 and k=5, the CDF is nearly 0.8. Since no MPH bin truly dominates and two similar tracks may have different spectra, the CDF does not approach 1.0 for any pair of (n, k) for $n < 30$ and $k < 10$.

4.3.2 Organization of OS-LSH Table

The result in Fig. 11 is used to design OS-LSH tables. A simple but practical procedure is shown in Fig. 12. For each MPH in the database, the positions of its k major MPH bins are determined (line 20), sorted in increasing order (line 21) and then used together as a hash key. The MPH is stored in the bucket associated with this hash key (line 22). With N audio tracks in the database and $\binom{n}{k}$ buckets, each bucket has on average $N/\binom{n}{k}$ MPHs.

4.3.3 Refined Probing Stage

Although similar audio tracks have many major MPH bins

Procedure storeMphInLshTable (MPH_i)
20. Sort the 66-D MPH_i in the decreasing order, let the original positions of top k major bins be $l_{i,j}$, j=1,2,…,k
21. Sort $<l_{i,j}>$ j=1,2,…,k in the increasing order as $<l'_{i,j}>$
22. Store MPH_i in the bucket associated with $<l'_{i,j}>$

Figure 12: Storage of MPHs in the OS-LSH table.

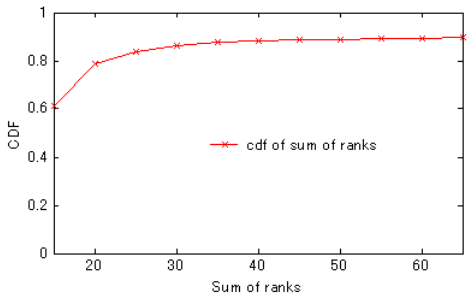

Figure 13: Cumulative density of sum of ranks.

in common, they do have differences. Multi-probing should be used to improve the recall. According to Fig. 11, the k major bins of a MPH appear in the n major MPH bins of its similar audio tracks with high probability. A simple but expensive method is to use all the k-subsets of the n major MPH bins of the query to form the hash keys, i.e. exhaustively probe all the possible $\binom{n}{k}$ buckets. But Fig. 11 also shows that as n increases, the increase in the CDF gets smaller and finally the CDF approaches a bound. So, most major MPH bins of an audio track appear in the top of the major bins of its relevant tracks and this fact can be used to refine the probing.

To refine the probing, we consider the following scenario: a query audio track Q is perceptually similar to an audio track R in the database and R can be found using exhaustive probing, i.e. the k major MPH bins of R are a subset of the n major MPH bins of Q. The ranks of R's k major MPH bins in the ordered list of the n major MPH bins of Q are recorded and the sum of the ranks is calculated. With n=15 and k=5, the minimal value of the sum of ranks is $\sum_{i=1}^{k} i = 15$, corresponding to the case where the k major MPH bins of R overlap with those of Q, although their orders may be different; the maximal value of the sum of ranks is $\sum_{i=n-k+1}^{n} i = 65$. Fig. 13 shows the probability of the sum of ranks in the range $[15, 65]$. Naturally, the sum of ranks has a small value for similar audio tracks and this is confirmed by the fact that the probability p(sum of ranks$\leq$25) is already high enough. Accordingly, among the buckets that potentially hold the similar audio tracks, only the buckets associated with k major bins whose sum of ranks is lower than a threshold should be probed.

Fig. 14 shows both recall and computation cost under different probe limits. The cost is the ratio of the number of MPHs compared in OS-LSH to that in kNN. As the probe limit increases, more buckets are probed. When the limit is above 25, the cost increases quickly while recall hardly changes. Accordingly, the probe limit is set to 25. By using this limit, the number of probed buckets is much less than $\binom{n}{k}$, which is the number of buckets probed when the probe limit is the maximal value of the sum of ranks (65).

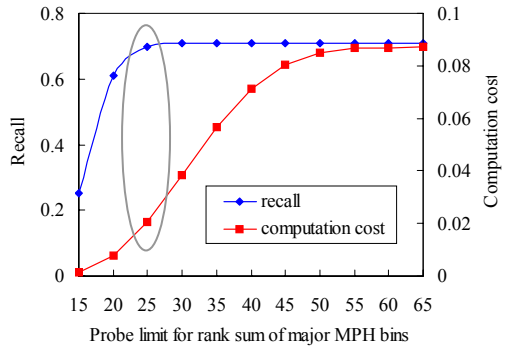

Figure 14: Recall and cost for different probe limits.

Procedure retrieveWithAnAudioQuery (q)
23. Calculate the MPH for q as MPH_q
24. Sort the 66-D MPH_q in the decreasing order, the original positions of top n major bins are $<l_{q,j}>$, j=1,2,…,n
25. From $<l_{q,j}>$, choose the sets of k major bins whose sum of ranks are no more than the probe limit
26. For each of the remaining sets of k major MPH bins
27. Sort positions of the k major bins in increasing order
28. Get MPHs in the bucket associated with these positions
29. Return those MPHs that are the kNN neighbors of MPH_q

Figure 15: Retrieval with an audio query.

4.3.4 *Overall Retrieval Process*

The retrieval process is shown in Fig. 15. Given an audio query q, MPH_q is computed (line 23) according to the procedure in Fig. 3. The positions of the n major bins of MPH_q are found (line 24). For each possible set of k major bins, its sum of ranks in the n major bins list is computed. Only the sets of k major bins whose sum of ranks is lower than the probe limit are kept (line 25). For each set, the positions of the k major bins are sorted in increasing order and used together as a hash key to locate the bucket (lines 26-28). All candidate MPHs in the associated buckets are compared against MPH_q and the relevant items are returned.

5. EVALUATION

In this section we describe our extensive experiments with live audio as queries and large test datasets to evaluate the performance of the approaches proposed above. We first describe the experimental setup, then provide the precision-recall results and study the effects of query length, environment noise and database size.

5.1 Experiment Setup

We employ the 5 datasets shown in Table 1, having a total of 30,396 audio tracks. Dataset I, Covers79, consists of 1072 covers of 79 songs. Datasets II, III, IV serve as background data. Dataset V is the noise. Datasets I, II and V are the same as in [7]. Since there is no large database publicly available for evaluating the scalability of audio retrieval, we collected audio tracks from the music channel of Youtube and included them in datasets III and IV.

In the experiments, each track is 30s long in mono-channel wave format, 16 bit/sample with sampling rate 22.05 KHz. The audio data is normalized, then divided into overlapping frames; a frame contains 1024 samples (46.4 ms) and the

Table 1: Dataset Description.

Datasets	Name	# Audio tracks
I	Covers79	1,072
II	ISMIR+RADIO+JPOP	4,203
III	Youtube I	5,850
IV	Youtube II	18,875
V	RNoise	396

adjacent frames have 50% overlap. Each frame is weighted by a Hamming window and padded with 1024 zeros. A 2048-point FFT is used to calculate the STFT from which the instantaneous frequencies are extracted and chroma features are obtained. Therefore, every audio track contains 1292 chroma features (12-dimensional). Through summarization, each track is compressed to a 66-dimensional MPH.

Unless stated otherwise, we employ the following setting: each of the 1072 tracks in Covers79 is used as the query to retrieve its relevant tracks from datasets I+II+III having 11,125 tracks; the exception is in the fourth experiment where dataset IV is used for evaluating the effect of database size. The query has the full length, like its relevant tracks, except in the second experiment where the effect of query length is measured. There is no extra noise except when evaluating the effect of noise in the third experiment. The number of ranked results equals that of relevant items, except for drawing the precision-recall curves.

We compare our proposal, MPH+OSLSH, to MPH with kNN (MPH+kNN) and to two other methods using global summarization: Pitch histogram with kNN (PH+kNN), feature union with kNN (FU+kNN) [2]. We also consider a method based on *local* summarization and multi-level LSH [7], LS+MLSH, in order to compare their respective balances between effectiveness and efficiency. The task is to detect and retrieve multiple relevant items with the query and rank them in an ordered list. We use recall, precision and F-measure [24] as metrics for algorithm effectiveness.

5.2 Precision-recall curves

We investigate here the precision-recall relationship for each of the five methods, by changing the number of retrieved items. The resulting recall-precision curves are shown in Fig. 16. MPH (MPH+kNN and MPH+OSLSH) outperforms both FU+kNN and PH+kNN. When precision equals 0.6, the recall reached by MPH+OSLSH is higher by 0.19 than that obtained with PH+kNN; LS+MLSH achieves a recall greater by 0.08 than that of MPH+OSLSH at the expense of post-sequence comparison. The degradation of recall for MPH+OSLSH shows that some information is inevitably lost during summarization and affects the discriminative power. But this is acceptable if we consider the significant improvement in storage and retrieval speed.

It is interesting to see that, for all methods, there are some critical recall values after which precision plummets (although the slope is a little different). Indeed, in the list of all audio tracks, most of the audio tracks relevant to the query appear near the top. The not-so-similar relevant tracks usually have higher differences to the query and are near the bottom of the list. They cannot be easily retrieved by simply increasing the number of outputs. Also, recall can hardly approach 1.0, especially when LSH is used. This is because the perceptually similar covers may have

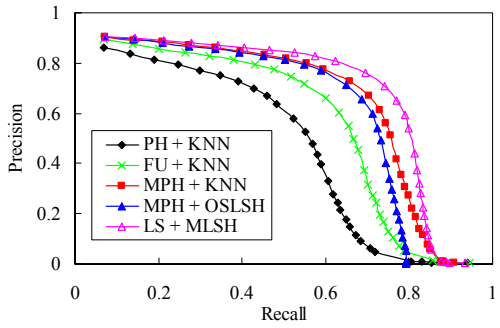

Figure 16: Precision vs. recall for all methods.

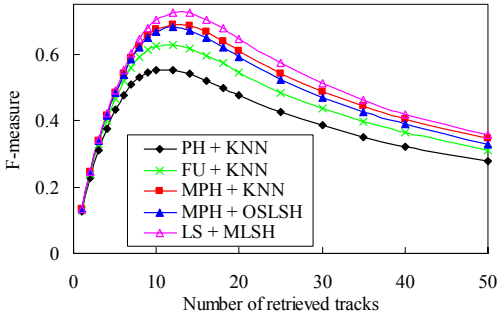

Figure 17: F-measure for the compared methods.

different spectral profiles and also because the discriminative capability of the audio sequence is diminished by the summarization. The recall of MPH+OSLSH remains nevertheless satisfactory.

The F-measure shown in Fig. 17 reflects the tradeoff between recall and precision. On average, a query has 12.5 relevant items in the database. Coincidently, the maximal F-measure is reached when 12 tracks are retrieved. MPH outperforms FU and PH. MPH+OSLSH approaches MPH+kNN and fills the gap between FU+kNN and LS+MLSH.

5.3 Effect of query length

A summary captures the statistics of an entire track. The audio signal is only short-term stationary, so the statistics change in time and the accuracy of a summary is affected by track length. It is necessary to measure the performance degradation when the query is shorter than its relevant tracks in the database.

Fig. 18 shows the recall obtained for different normalized query lengths. Since the accuracy of a summary decreases with query length, the recall decreases as well. But for MPH the decrease in recall is relatively slow when the query length is greater than 0.5. This is due to the fact that, although music signal is not technically stationary, part of it is usually repeated because of the score composition. In this sense, even a short sequence carries most of the typical frequencies of the entire audio track, so the accuracy of the summary can be relatively well preserved.

Furthermore, since MPH retains information about temporal correlations, it is more discriminative than PH, so it shows a much higher recall. FU relies on averages and standard deviations for computing the summary, so its performance more strongly depends on query length.

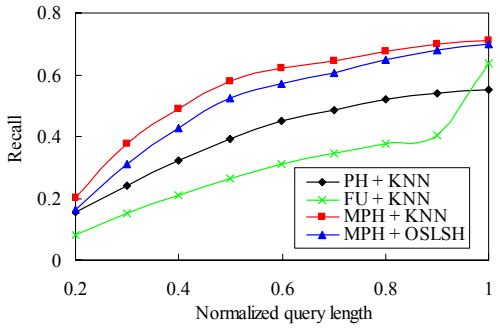

Figure 18: Recall for different query lengths.

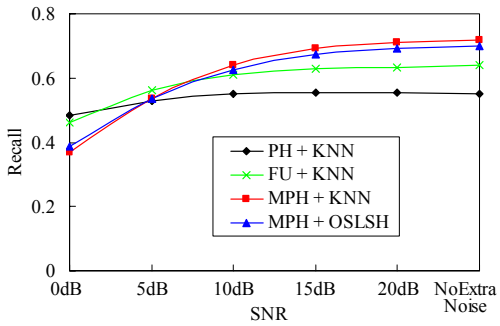

Figure 19: Recall under different SNR.

5.4 Effect of environmental noise

A potential application of query-by-audio music retrieval is finding the original track after recording music in a natural environment. In such a scenario the recorded query is likely to be affected by non-white environmental noise and its signal to noise ratio (SNR) can be low.

To investigate the effect of noise, 396 queries are randomly selected from Covers79 and combined with the 396 different noise segments of the RNoise dataset, at several values of the SNR, to simulate a real noisy environment. Fig. 19 shows that at very low SNR the pitch histogram and MPH are affected by the non-white noise. FU, keeping more information, performs a little better. But when SNR is above 10dB, the degradation of the recall is small, and MPH+kNN outperforms PH+kNN and FU+kNN. MPH+OSLSH performs almost as well as MPH+kNN.

5.5 Effect of database size

Using an index is especially important for large databases. By varying the database size from 5,000 to 30,000, we evaluate the recall and computation cost of the different methods. As shown in Fig. 20, for all the methods recall decreases as the database size increases. However, the difference between MPH+OSLSH and MPH+kNN slightly diminishes while the difference between MPH+OSLSH and PH+kNN or FU+kNN slightly increases.

Fig. 21 shows the ratio of the number of MPHs compared with MPH+OSLSH to that with MPH+kNN. As the database size increases, the normalized computation cost decreases, which confirms the scalability of the proposed LSH scheme based on order statistics. When there are 30,000 audio tracks in the database, MPH+OSLSH achieves a recall of 0.671 and retrieval is 58.8 times faster than with MPH+kNN.

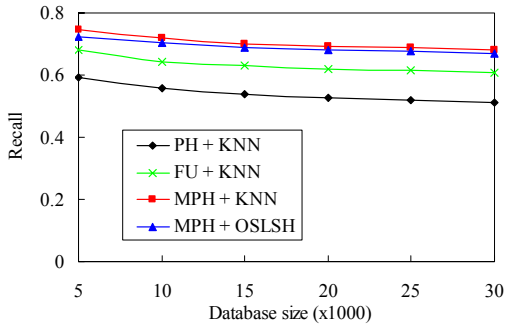

Figure 20: Recall for different database sizes.

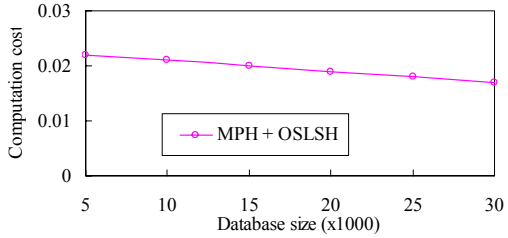

Figure 21: Relative cost MPH+OSLSH vs. MPH+kNN.

5.6 Comparison of retrieval methods

A comparison among methods, based on the theoretical analysis (for N audio tracks) and on the experimental results (on the dataset with N=11,125 tracks), is given in Table 2. Three aspects are considered: (i) Storage. PH, FU and MPH use global summarization and their feature dimensions are 88, 218 and 66 respectively. LS+MLSH uses local summarization and the dimension depends on track length; for a 30s track 1292 chroma features are generated and with a compression ratio δ=0.15 [7] the number of local summaries is $d = 1292 \cdot 0.15 = 194$. (ii) Cost of retrieval. With a global summary (FU, PH, MPH) and kNN the computation cost is determined by the storage. MPH+OSLSH accelerates pairwise comparisons by a factor α_1 with respect to MPH+kNN. For LS+MLSH the total number of pairs is $N \cdot d^2$, pairwise comparisons are accelerated by α_2, and the *a posteriori* sequence comparison is performed via the Hough transform (whose cost is denoted by H). Even without considering the sequence comparison in LS+MLSH, MPH+OSLSH achieves a significant speedup estimated by $(N \cdot d^2 \cdot 12/\alpha_2)/(66 \cdot N/\alpha_1)$. As d augments with the length of the audio tracks, this speedup further increases. In the experiments we found $\alpha_1 = 47.6$ and $\alpha_2 = 61.2$. The relative computation cost of MPH-OSLSH with respect to MPH+kNN is 0.021, while the relative cost of LS+MLSH with respect to MPH+kNN is 41.9 (pairwise comparison) + 22.7 (*a posteriori* comparison). Therefore, retrieval is 3076 times faster with MPH+OSLSH than with LS+MLSH. (iii) Recall. MPH+OSLSH has a satisfactory recall, much higher than those of FU+kNN and PH+kNN, although a little less than that of LS+MLSH. In all, MPH+OSLSH achieves the best balance among the key factors: storage, computation cost and recall.

6. CONCLUSION

Scalable content-based audio retrieval and mining requires efficient similarity search for acoustic feature sequences. Re-

Table 2: Comparison of retrieval schemes.

	Theoretical estimates		Experiment results	
Method	Storage	Computation	Computation	Recall
PH+kNN	$88{\cdot}N$	$88 \cdot N$	1.33	0.550
FU+kNN	$218{\cdot}N$	$218 \cdot N$	3.30	0.638
MPH+kNN	$66{\cdot}N$	$66 \cdot N$	1.00	0.712
MPH+OSLSH	$66{\cdot}N$	$66 \cdot N/\alpha_1$	0.021	0.700
LS+MLSH	$12{\cdot}dN$	$Nd^2(12+H)/\alpha_2$	64.6	0.781

cent research has shown that LSH techniques can effectively filter out non-similar features and speed up the search. But this does not completely solve the problem when similar *sequences* of features have to be retrieved. In this paper we proposed Multi-Probe Histograms as global summaries of audio feature sequences that retain local temporal acoustic correlations by concatenating major bins of adjacent chroma features. Based on an analysis of the characteristics of multi-probe histograms we also exploited their order statistics to more efficiently organize and probe LSH tables. The resulting approximate retrieval method is comparable in accuracy with exact kNN retrieval, while showing a significantly faster retrieval speed. Experimental evaluations on large datasets with up to 30,000 audio tracks confirmed the robustness and effectiveness of the proposed algorithms. Finally, we can point out that the principle of multi-probe histograms can be directly applied to other multimedia retrieval and mining applications.

7. ACKNOWLEDGMENTS

This work was supported in part by a grant from DoD-ARL through the KIMCOE center of Excellence, Fondazione Cariplo 2007 (grant Capitale Umano di Eccellenza) and by Agence Nationale de la Recherche (grant ANR-07-MDCO-017).

8. REFERENCES

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

[2] Y. Yu, K. Joe, V. Oria, F. Moerchen, J. Stephen Downie, and L. Chen. Multi-version music search using acoustic feature union and exact soft mapping. *Int. Journal of Semantic Computing*, 3(2):209–234, 2009.

[3] M. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional spaces. *IEEE Trans. Audio, Speech and Lang*, 16(8):1015–1028, 2008.

[4] 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.

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

[6] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th ACM STOC*, 1998.

[7] Y. Yu, M. Crucianu, V. Oria, and L. Chen. Local summarization and multi-level LSH for retrieving multi-variant audio tracks. In *Proc. ACM MM'09*, pages 341–350, 2009.

[8] M. Muller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proc. ISMIR'05*, pages 288–295, 2005.

[9] M. Muller, S. Ewert, and S. Kreuzer. Making chroma features more robust to timbre changes. In *Proc. ICASSP'09*, pages 1869–1872, 2009.

[10] 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.

[11] G. Chechik, E. Le, M. Rehn, S. Bengio, and D. Lyon. Large-scale content-based audio retrieval from text queries. In *Proc. MIR'08*, pages 105–112, 2008.

[12] J. S. Downie. The music information retrieval evaluation exchange (MIREX). *D-Lib Mag.*, 12(12), 2006.

[13] M. Robine, P. Hanna, P. Ferraro, and J. Allali. Adaptation of string matching algorithms for identification of near-duplicate music documents. In *Proc. SIGIR Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, 2007.

[14] J. F. Serrano and J. M. Inesta. Music motive extraction through hanson intervallic analysis. In *Proc. CIC'06*, pages 154–160, 2006.

[15] M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio retrieval. In *Proc. ISMIR'08*, pages 295–300, 2008.

[16] R. Miotto and N. Orio. A music identification system based on chroma indexing and statistical modeling. In *Proc. ISMIR'08*, pages 301–306, 2008.

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

[18] A. Joly and O. Buisson. A posteriori multi-probe locality sensitive hashing. In *Proc. ACM MM'08*, pages 209–218, 2008.

[19] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. Neural Information Processing Systems*, 2008.

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

[21] W. Wang and S. Wang. A scalable content-based image retrieval scheme using locality-sensitive hashing. In *Proc. Int. Conference on Computational Intelligence and Natural Computing*, pages 151–154, 2009.

[22] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *Proc. ACM MM'08*, pages 61–70, 2008.

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

[24] http://en.wikipedia.org/wiki/information retrieval.