Audio Retrieval Based on Perceptual Similarity

Teng Zhang, Ji Wu
Department of Electronic Engineering
Tsinghua University
Beijing, China
Email: {zt1887@126.com,wuji_ee@mail.tsinghua.edu.cn}

Dingding Wang
Department of CEECS
Florida Atlantic University
Boca Raton, FL, USA
Email: wangd@fau.edu

Tao Li
School of Computer Science
Florida International University
Miami, FL, USA
Email: taoli@cs.fiu.edu

***Abstract*—Given a short query audio clip, the goal of audio retrieval is to automatically fetch all similar clips from a given audio database. Different from traditional audio similarity which is mainly based on priori knowledge of objective reality, this paper proposes to use a more subjective method to measure the perceptual similarity between audio clips. These perceptual features focus on users' personal experience, which can be very helpful for audio retrieval across different databases. In addition, indexing and audio matching methods are introduced to speed up the retrieval process. Experimental results on four different datasets are conducted to evaluate the effectiveness and efficiency of our proposed approaches.**

***Keywords*-audio retrieval; perceptual similarity**

I. Introduction

Content-based audio retrieval has been a challenging task for a long time. Given a short query audio clip, the goal of audio retrieval is to automatically retrieve all similar clips from a given audio database. Hence the essential problem is to calculate the similarity between audio clips, which is particularly difficult for waveform inputs. The difficulty mainly relies on the following aspects. First, the waveform audio data is too complex for computer to process directly. To deal with this problem, a general method is to convert the audio data within the database into sequences of suitable audio features. Second, researchers notice that the feature-based similarities between audio clips are not as intuitive or user-friendly as they had expected, which is often referred as the problem of "bridging the semantic gap" [1]. Finally, the retrieval process is often time-consuming and often can not meet the requirement of real-time retrieval.

In this paper, we use a natural strategy to conduct audio retrieval. First of all we use some common features to represent audio effectively; then we test the effectiveness of various acoustic features in our perceptual similarity measurement and select suitable features to represent the perceptual similarity between audio clips; finally we introduce the indexing and audio matching method to speed up the retrieval process.

There are some common features used in many audio tasks including Zero Crossing Rate (ZCR), Mel-frequency Cepstral Coefficients ($MFCC$), Line Spectral Pairs (LSP), etc. Johnson et al. [3] used Mel PLP cepstrum coefficients of the audio and a covariance-based distance metric to quickly locate audio repeats. Muscle Fish system [2] used pitch, timbre, loudness and brightness to represent the audio. Muller et al. [4] proposed a new type of chroma-based feature that strongly correlated to the harmonic progression of the audio. Melih et al. [5] showed that a new structured representation of audio features was helpful for content-based audio retrieval. Many research efforts have also been reported on feature extraction methods for music audio data [19], [20].

However, traditional signal based audio features are often directly extracted from the audio, which makes the audio retrieval task into a simple audio lookup problem [15], [16], [17], [20]. As a result, only the same homologous audio can be recalled, which leads to a high precision but poor recall rate. In recent years, semantic information is emerging in the field of audio retrieval. Kurth et al. [8] defined the melody similarity to deal with the music variance. Barrington et al. [10] trained several SVM (Support Vector Machine) classifiers for semantic tags to improve the retrieval performance. The definition of "human-centered" similarity is becoming an important exploration direction of audio retrieval. In our task, a novel definition of perceptual similarity is proposed to guide feature extraction and similarity calculation. The new perceptual similarity shows the potential to break through the limit of traditional methods.

The audio retrieval efficiency is mainly determined by two factors: the computational complexity of audio similarity and the search strategy. Bosteels et al. [6] proposed a fuzzy similarity calculation based on spectrum histograms and fluctuation patterns. Lo et al. [7] divided the audio into several homogeneous segments, and trained an ensemble classifier to provide the audio annotation. Kurth et al. [8] employed standard indexing techniques to obtain an efficient index-based audio matching procedure. Zhang et al. [9] took two stages to speed up the retrieval process, which consisted of coarse search based on the histogram pruning algorithm and retrieval based on time information. In this paper, we present an indexing method and a multi-stage matching procedure to speed up the retrieval process. Experimental results on four different datasets are conducted to evaluate the effectiveness and efficiency of our proposed approaches.

This paper is organized as follows. Section II illuminates the perceptual similarity and discusses a feature selection procedure. Section III introduces the indexing method and provides feasible implementation methods. Section IV uses a Multi-stage matching strategy to complete the audio retrieval task and discusses the time efficiency. Section V conducts

COLLABORATECOM 2014, October 22-25, Miami, United States

DOI 10.4108/icst.collaboratecom.2014.257704

experiments and evaluates the performance of the proposed system. Section VI concludes this paper and presents our future work.

II. Perceptual Similarity

Given an audio clip pair, there are many distance definitions to evaluate their similarity. Traditional feature-based similarity only conveys raw signal information, but ignores the perceptual information that the audio transmits to people. It is hard to define perceptual similarity between audio clips because of the variance among people and also the variance related to external conditions. It is a process of a subjective judgment more or less. Thus we introduce MOS (Mean Opinion Score [14]), a scoring method of Voice Quality Test to evaluate the perceptual similarity between audio clips.

The perceptual similarity degree is experimentally divided into four scales as shown in Table I, and we use the statistical average scores from a number of people as the subjective assessment of the perceptual similarity. The larger the score is, the more similar the audio pair is.

The measurement of the perceptual similarity provides a proper link between the acoustic features and the perceptual similarity. Given a specific acoustic feature, we can calculate the acoustic distance as an objective measurement of the perceptual similarity. If the acoustic distance well correlates with the perceptual similarity, the acoustic feature is considered to be effective. However, one single feature may not be that effective, so we propose to find a complementary and effective integration of acoustic features via a feature selection procedure.

In this paper, we use the $SFFS$ (Sequential Floating Forward Selection) procedure [12] to greedily select the optimal features. Assume that there are N kinds of acoustic features, denoted as $S = \{s_1, s_2, ..., s_N\}$, and correspondingly there are N kinds of distances, denoted as $D = \{d_1, d_2, ..., d_N\}$.

Finally, we get a set of acoustic features U that is considered most effective for the measurement of perceptual similarity.

III. Indexing Methods

The time cost of an audio matching method linearly depends on the size of the database. Kurth et al. [8] used the indexing method to speed up the retrieval process. Here we introduce the indexing idea from unsupervised learning, discuss its limitations, and then propose the ameliorated indexing procedure.

Recall that U is the feature set we have selected to represent the perceptual similarity of audio clips. If we can find a finite set $C = \{c_1, c_2, ..., c_K\}$, where each index $i \in [1 : K]$ corresponds to a feature class determined by a nearest neighbor criterion, C is called the codebook of U. A common strategy to find C is based on unsupervised learning algorithms such as clustering. In this paper, we use the known $ISODATA$ (Iterative Self-Organizing Data Analysis Technique Algorithm) [13] to obtain the codebook C.

When we use the codebook selected by the nearest neighbor criterion to execute the audio retrieval process, we often make the following assumptions: (1) samples in the same class are more similar (**Figure 1(a)**); (2) samples in different classes are different (**Figure 1(b)**); and (3) similar audio clips have the same length.

However in Figure 1(a), A and B are both assigned to C_1, but obviously they are not similar. In Figure 1(b), A and B are assigned to different classes, but they are similar based on Euclidean distance. Thus we make the following corrections to the original indexing method.

1) **C1.** Assume that the distance between A (or B) and class center C_1 is d_AC (or d_BC). If $\mid d_AC - d_BC \mid > \theta_s$, A and B are not similar; Otherwise, use the nearest neighbor criterion to provide a judgement.

Algorithm 1 SFFS Feature Selection Procedure.

Input: The set of acoustic features, $S = \{s_1, s_2, ..., s_N\}$; MOS perceptual similarity scores of all audio pairs, $score$;
Output: Selected feature integration, U;

1: Initialization: Select the initial feature integration $U = \{s_{1'}, s_{2'}, ..., s_{k'}\}$. The residual set is $L = S - D$;
2: Calculate the cross-validation classification accuracy $cvAcc$ on U via SVM;
3: Forward process: For each feature in L, add int to U and calculate its $cvAcc$. If none of the addition improves the accuracy, goto **4**, else, add the feature which mostly improves the accuracy to U, and repeat **3**;
4: Backward process: If the size of U is bigger than one, exclude each feature in U and calculate its $cvAcc$, If none of the exclusion improves the accuracy, goto **5**, else, exclude the feature which mostly improves the accuracy from U, and repeat **4**;
5: If the accuracy is unchanged, goto **6**, else, update $L = S - D$, goto **3**;
6: **return** U;

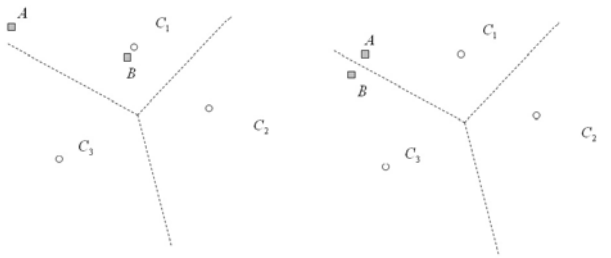
Algorithm 2 ISODATA Procedure.

Input: Features related to perceptual similarity, U; Expected class number, K; Least samples in class, θ_k; Discrete degree in class, θ_s; Discrete degree between classes, θ_c; Initial class number, N_c;
Output: Selected codebook, C;

1: Initialization: Select the initial class centre $C = \{c_1, c_2, ..., c_{N_c}\}$;
2: Delete: Allot samples U into different class C by a nearest neighbor criterion, delete the class whose sample number is under θ_k, update C;
3: Split: Calculate discrete degree in each class, split the class whose discrete degree is bigger than θ_s into two new classes, update C;
4: Merge: Calculate discrete degree between classes, Merge the classes whose discrete degree is smaller than θ_c into one class, update C;
5: Iterate: According to different conditions, iterate the Step 2-4 until all conditions are satisfied;
6: **return** C;

TABLE I: **perceptual similarity definitions**

MOS	level of similarity	description
4	almost the same	the difference is slight, difficult to detect
3	similar	there are small differences, relatively obvious
2	a little similar	there are many differences, but some similarity
1	not similar at all	obvious differences, easy to distinguish



(a) same class (b) different classes

Fig. 1: A counter example.

2) **C2.** Instead of classifying each point to one exact class, we provide $nbest$ labels corresponding to the nearest classes. If the label lists of two points intersect with at least one label, we say that the tow points may be similar.
3) **C3.** We use a limited flexible method instead of the complex DTW (Dynamic Time Warping [11]) technique to solve the unequal-length series matching problem.

When we have a query audio Q and an audio clip D_k within the database, we can calculate their similarity using the following formula:

$$S(Q, D_k) = \sum_{j=0}^{L_Q - 1} Score(j, k), \tag{1}$$

where $Score(j, k)$ indicates the matching score of $Q(j)$ in D_k. From C3, $Score(j, k)$ can be calculated as follows,

$$Score(j, k) = \max_{i=-N:N} \{c_i \cdot s(Q(j), D_k(j + i))\}. \tag{2}$$

Assuming that the label list of $Q(j)$ is $s_j = \{l_1, l_2, ..., l_{nbest}\}$ and the label list of $D_k(i)$ is $b_i = \{q_1, q_2, ..., q_n best\}$, from C1 and C2 the definition of $s(Q(j), D_k(i))$ can be given as follows,

- Assuming that $C = s_j \cap b_i = \{p_1, p_2, ..., p_N C\}$, and $d_i = \mid d_{sp} - d_{bp} \mid$, we can define $s(Q(j), D_k(i))$ as follows,

$$s_1(Q(j), D_k(i)) = \begin{cases} 1 \quad if\ C \neq \phi\ and\ \exists i \in [1, NC] \\ \ s.t.\ d_i < \theta_s; \\ 0 \quad else. \end{cases} \tag{3}$$

- If the $NC > 1$, we can furthermore get a better definition,

$$s_2(Q(j), D_k(i)) = \sum_{i=1}^{NC} a_i \cdot I_{(d_i < \theta_s)}. \tag{4}$$

The final audio indexing procedure is presented as **Algorithm 3**.

Algorithm 3 The Audio Indexing Procedure.

Input: Audio Database, D; Query audio clip, Q; Similarity threshold, T_sim;
Output: Candidate similar audio clips, E;

1: Index construction: Extract perceptual features of D and Q as **Algorithm 1**, select the codebook C of all frames in D using **Algorithm 2**, and construct the database index with the $nbest$ classes and each class i is labeled as $L(i) = \{i_1, i_2, ...\}$, where i_k is the frame number corresponding to i;
2: Query process: Classify the query clip to $nbest$ classes using the nearest neighbor criterion, and get the query label list $Q(j) = s_j = \{l_1, l_2, ..., l_{nbest}\}$ and the candidate matching frames $L(s_0), L(s_1), ..., L(s_{L_Q - 1})$;
3: Locate the matching frames: Calculate the similarity between Q and candidate audio clip D_k which can be determined by L using **formula (1)**;
4: Candidate selection: If $S(Q, D_k) > T_sim$, add D_k to the candidate set E;
5: **return** E;

IV. Multi-stage Matching

Although the indexing method has greatly reduced the audio quantity, the amount of the candidate similar audio clips is still too large. A general strategy for audio matching is to construct a probabilistic model to characterize the distribution of audio features, and then calculate their similarity by measuring the difference between the models. Wold et al. [2] counted the mean and variance of audio features to perform the audio retrieval task. Another useful method is DTW, which can align two sequences with different lengths into the same length.

Assume that we have two audio sequences $S = \{s_1, s_2, ..., s_{ls}\}$ and $T = \{t_1, t_2, ..., t_{lt}\}$, and their length are ls and lt, respectively. In the DTW algorithm, the start and end points of S and T are forced aligned. If the track found in Figure 2 is the best matching track of S and T, the track from the origin to K is obviously the best matching track of $S^{'} = \{s_1, s_2, ..., s_K\}$ and $T^{'} = \{t_1, t_2, ..., t_K\}$. Now assuming that the distance from the origin to K is $D(i, j)$, the similarity between s_i and t_j becomes d_{ij}, then we have the recursion

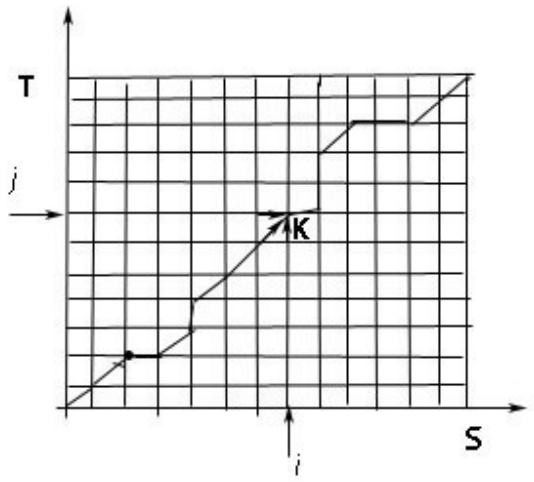


Fig. 2: The DTW algorithm.

formula as follows,

$$\begin{cases} D(i,j) = \min\{D(i-1,j) + w_1 \cdot d_{ij}, D(i-1,j-1) \\ \quad + w_2 \cdot d_{ij}, D(i,j-1) + w_3 \cdot d_{ij}\}; \\ D(1,1) = d_{11}; \\ D(1,j) = D(1,j-1) + w_3 \cdot d_{1j} \quad j > 2; \\ D(i,1) = D(i-1,1) + w_1 \cdot d_{i1} \quad i > 2. \end{cases} \tag{5}$$

In this paper, we propose a multi-stage matching procedure to speed up the retrieval process. The first step is to calculate the mean of audio features, and set a threshold to exclude some dissimilar clips. Then the $MFCC$ distance using the DTW algorithm is introduced to complete the rough matching of the query audio. Finally, a SVM classifier based on the perceptual distance introduced in Section II is used to generate a refined matching result.

According to the discussion in Section II, we select the $MFCC$ and LSP features to calculate audio distances. As shown in Figure 3, the mean distance of $MFCC$ and the distance of $MFCC$can somehow distinguish similar audio clips, but on the other hand, the overlap of similar and dissimilar samples will lead to a high loss rate.

The calculation of the mean of features is fast but not accurate, so we introduce a less efficient but more accurate DTW method to measure audio distances. Figure 4 shows the $MFCC$ distance using the DTW algorithm. We can see that the overlap is reduced, which means a lower loss rate as a result.

In Section II, we have introduced an integration of acoustic features $U = \{u_1, u_2, ..., u_k\}$ that is considered most effective for the measurement of perceptual similarity. Using the DTW algorithm for each feature, we get k distance measure $D = \{d_1, d_2, ..., d_k\}$, then we can use the SVM classifier to generate the final result.

V. Experiments

We conduct various experiments to evaluate our audio retrieval system. We first introduce the data sets we used in our experiments (Section V.A). Prior to evaluating the performance and efficiency of our system, we give the result of feature selection based on our definition of perceptual similarity (Section V.B). Then we investigate the indexing method and its effectiveness (Section V.C). Finally, as the main result of this paper, we examine the overall performance and efficiency of the entire system (Section V.D).

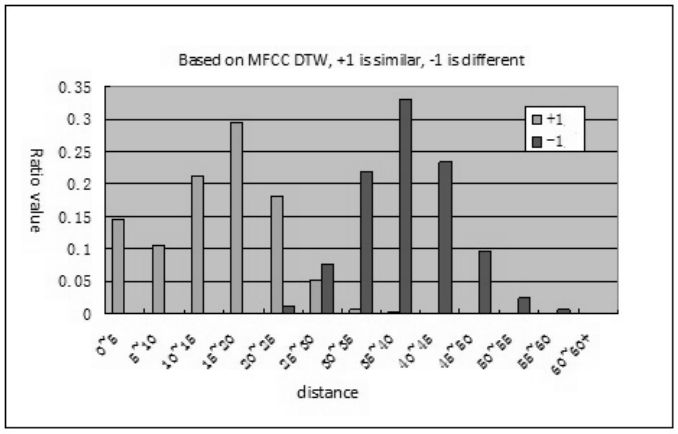


Fig. 4: The $MFCC$ distance using the DTW algorithm.

A. Data Sets

There are four data sets used in our experiments. The audio clips in the same set come in pairs.

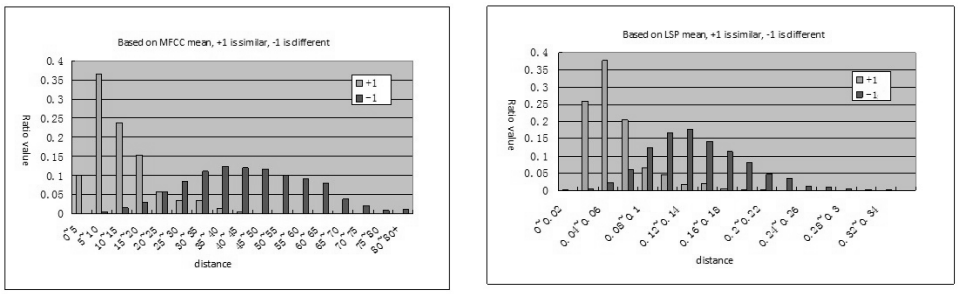
- $863Data$ is reading-style Chinese speech, and is supported by the 863 program. It includes 271 pairs of clips.
- $Switchboard$ is Chinese speech from free-style phone conversations. There are 533 audio pairs in it.
- $BNBC$ is $CCTV$ News broadcast, and includes 2375 pairs of clips.
- $Songs$ is a collection of Chinese Web songs, and includes 43 pairs of clips.

B. Feature Selection

Here we use all the data to test the effectiveness of acoustic features for the perceptual similarity we have defined in Section II. The acoustic distance related to each feature is calculated using the DTW algorithm as described in Section IV. The correlation coefficient ρ_i between feature s_i and the perceptual similarity is calculated and compared in Table II. Note that the larger ρ_i is, the more effective s_i is.

Each single feature in Table II shows a corresponding correlation to the perceptual similarity, but as described in Section II, we need to find a complementary and effective integration of acoustic features using the $SFFS$ procedure.

We use part of $BNBC$ data and all the $Songs$ data for the feature selection. All the acoustic features in Table I are used for the $SFFS$ procedure. The initial feature sets and the selected optimal sets are shown in Table III. U_1 is the LSP feature set, U_2 is the $MFCC$ feature set, and U_3 is the PLP feature set. They are the most effective features according to Table II.



(a) MFCC mean (b) LSP mean

Fig. 3: Distances based on $MFCC$ and LSP features.

TABLE II: **Correlation coefficient ρ_i between feature s_i and perceptual similarity.**

Feature	ρ_i	Feature	ρ_i	Feature	ρ_i
pitch	-0.42	pitch delta	-0.13	energy	-0.33
energy delta	-0.54	chroma	-0.55	intensity	-0.17
intensity delta	-0.19	loudness	-0.28	loudness delta	-0.31
LSP	-0.72	LSP delta	-0.63	ZCR	-0.45
ZCR delta	-0.58	MFCC	-0.73	MFCC delta	-0.61
PLP	-0.74	PLP delta	-0.61	centroid	-0.52
centroid delta	-0.49	entropy	-0.46	entropy delta	-0.44
flux	-0.23	flux delta	-0.33	roll off-0.25	-0.51
roll off-0.50	-0.51	roll off-0.75	-0.55	roll off-0.90	-0.53

TABLE III: **The initial feature sets and the selected optimal sets.**

Initial set (feature index)	Selected set (feature index)
$U_1 = \{10, 11\}$	$S_1 = \{5, 9, 10, 12, 16, 21, 23\}$
$U_2 = \{14, 15\}$	$S_2 = \{2, 8, 9, 14, 15\}$
$U_3 = \{16, 17\}$	

Using the feature sets mentioned above, we conduct an experiment to determine whether a given audio pair is perceptually similar or dissimilar. 60% of all the data are used as training data and the rest are testing data, the result of the similarity classification is represented in Table IV. We can see that the selected set S_1 gives the best accuracy and an acceptable recall rate, so the features in S_1 will be used for the measure of perceptual similarity.

C. Audio Indexing

Before conducting the indexing experiment, the parameters are listed in Table V. In this set of experiments, we examine the impact of $nbestD$, $nbestQ$, $nCluster$ and NT on the performance of the indexing procedure.

TABLE IV: **The results of similarity classification with different feature sets.**

Feature set	Accuracy (%)	Recall rate (%)	False alarm rate (%)
U_1	84.45	84.34	15.52
U_2	87.09	78.65	10.55
U_3	86.86	82.92	12.04
S_1	**89.04**	81.49	**8.86**
S_2	86.24	82.21	12.64

TABLE V: **Parameters related to indexing.**

parameters	description
$nCluster$	cluster number
$nbestD$	database index number
$nbestQ$	query index number
NT	matching radius
T_{dis}	the largest allowed distance between similar audio clips
T_{FRR}	the maximum allowable miss rate during indexing
$FAR_{T_{FRR}}$	the false alarm rate when T_{FRR}
ROC	the area under curve FA-FR
$Time$	indexing time

From Table VI and Table VII, we have three observations as follows.

- When $(nbestD, nbestQ) = (3, 2)$, $FAR_{0.05}$ performs the best. If $nbestD/nbestQ$ becomes larger, the false alarm rate will be higher. If $nbestD/nbestQ$ becomes smaller, the miss rate will be higher.
- When $NT = 1$, $FAR_{0.05}$ and ROC perform the best, but the time cost is a little larger than $NT = 0$.
- When $nCluster = 64$, $FAR_{0.05}$ and ROC perform better than $nCluster = 128$, but the time cost is a little larger.

D. Audio Matching

As mentioned in Section IV, we propose a multi-stage matching procedure to speed up the retrieval process. The whole system described in this paper is shown in Figure 5, where $stage0$ is the indexing procedure, $stage1$ represents the distance based on the mean of acoustic features ($MFCC$ and LSP), $stage2$ uses the DTW algorithm to calculate the distance based on single $MFCC$ features, and $stage3$ uses the feature set we have selected in Section V.B to calculate the distance.

We use $863Data$, $Switchboard$, part of $BNBC$, and $Songs$ as training set, the rest of $BNBC$ as testing set to examine the performance of our system. The performance of each matching stage in the training set is shown in Table VIII, and the performance of the entire system in the testing set is shown in Table IX. The time cost of each stage is listed in Table X.

During the multi-stage matching procedure, each stage will get rid of dissimilar candidates, thus the DTW distance based

TABLE VI: **The change trend of $FAR_{T_{FRR}}$ along with $nbestD$ and $nbestQ$.**

$(nbestD, nbestQ)$	$FAR_{0.02}$(%)	$FAR_{0.05}$(%)	$FAR_{0.08}$(%)	$FAR_{0.10}$(%)
$(2, 2)$	3.3758	1.6539	1.1397	0.9199
$(2, 3)$	3.7527	1.4038	1.0641	0.7650
$(3, 2)$	**2.2301**	**1.0620**	0.8103	0.7109
$(3, 3)$	5.8140	1.2345	**0.4256**	**0.3588**

TABLE VII: **The indexing performance along with NT and $nCluster$, when $(nbestD, nbestQ) = (3, 2)$.**

performance	$nCluster$	NT			
		0	1	2	3
$FAR_{0.05}(\%)$	64	1.5166	**0.9478**	1.0620	1.2267
	128	2.6380	1.8267	1.9177	2.4863
ROC	64	0.00917	**0.00905**	0.00953	0.00988
	128	0.00922	0.00965	0.01017	0.01094
$Time(s)$	64	1.42118	1.66252	1.91380	2.05800
	128	**1.29132**	1.38518	1.50084	1.56142

TABLE VIII: **Stage performance in training set.**

stage	accuracy(%)	recall rate(%)	false alarm rate(%)
mean based distance	84.98	63.48	9.28
$MFCC$ based DTW distance	87.83	74.96	8.73
multiple features based DTW distance	**89.82**	**83.36**	**8.45**

TABLE IX: **System performance in testing set.**

	accuracy(%)	recall rate(%)	F-measure(%)
audio	67.7	90.1	77.3
speech	91.5	83.3	87.2

TABLE X: **Time cost of each stage (database length = 17368.42s).**

query length	retrieval time(s)	stage 0(s)	stage 1(s)	stage 2(s)	stage 3(s)
17.353	0.468	0.203	0	0.016	0.124
3.25	0.109	0.047	0	0	0
2.707	0.14	0.063	0.016	0	0
27.312	0.89	0.281	0	0.077	0.454
2.565	0.125	0.031	0	0.016	0
1.998	0.093	0.031	0	0	0
2.54	0.109	0.031	0	0	0
9.351	0.218	0.11	0	0.031	0.016
16.961	2.046	0.172	0	0.188	1.61
3.668	0.109	0.047	0	0	0
4.073	0.14	0.047	0	0.015	0.031
2.539	0.109	0.031	0	0	0
2.539	0.109	0.031	0	0	0.016
2.738	0.109	0.031	0	0	0
7.184	0.156	0.078	0	0.015	0
19.579	1.64	0.188	0	0.077	1.282
8.409	0.187	0.094	0	0	0.016
134.766	6.757	1.516	0.016	0.435	3.549

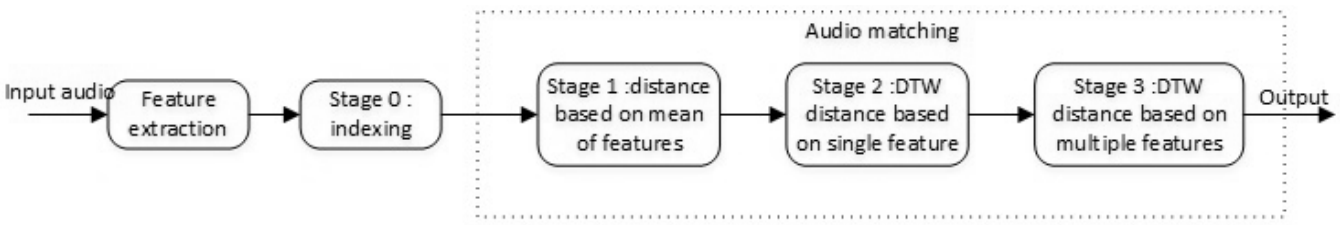


Fig. 5: The proposed system.

on multiple features shows the best performance in Table VIII. When the query audio clip contains only speech (Table IX), the performance of our system will become even better, since we mainly use speech data to select perceptual features and get the indexing result. Table X shows that stage 3 takes about 52% of the total time cost, which means that we can properly set the threshold in $stage1$ and $stage2$ to balance the performance and the time cost.

VI. Conclusion

In this paper, we proposed a novel audio retrieval method based on perceptual similarity, which provides some clues to the relevance between acoustic audio features and the semantic information. One simple yet important idea is to define the perceptual similarity between audio clips, which helps us to select the most effective feature set. Then we introduce an indexing method and a multi-stage matching procedure to speed up the retrieval process. Finally, we obtain 0.667 in accuracy and 0.901 in recall rate using the audio data in the experiments. When the query is limited to speech, our accuracy goes up to 0.915 and our recall rate descends to 0.833. The retrieval time of a query clip with the length of 134.766 seconds in a database with the length of 17368.42 seconds is only 6.757 seconds, which satisfies the requirement of real-time retrieval.

However, the performance of our system in music-related retrieval task is not good, which needs more efforts to expand the coverage of our system to all types of audio.

VII. Acknowledgements

The work of T. Zhang and J. Wu is partially supported by the National Natural Science Funds of China under Grant 61170197, the Sub-Project of 863 Hi-Tech Key Project under Grant 2012AA011004, and the Planned Science and Technology Project of Tsinghua University under Grant 20111081023. The work of D. Wang and T. Li is partially supported by the National Science Foundation under grants DBI-0850203, CNS-1126619 and IIS-1213026 and by the U.S. Department of Homeland Security under Grant Award Number 2009-ST-061-CI0001-06.

References

[1] Michael S. Lew, Nicu Sebe, Chabane Djeraba, Ramesh Jain. Content-based multimedia information retrieval: state of the art and challenges. In ACM Transactions on Multimedia Computing, Communications and Applications, vol.2(1): 1-19, 2006.

[2] E. Wold, T. Blum, D. Keislar, J. Wheaten. Content-based classification, search, and retrieval of audio. In IEEE Multimedia, vol.3(3): 27-36, 1996.

[3] Sue E. Johnson, Philip C. Woodland. A method for direct audio search with applications to indexing and retrieval. In Proceedings of ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3: 1427-1430, 2000.

[4] Meinard Muller, Frank Kurth, Michael Clausen. Audio Matching via chroma-based statistical features. In Proceedings of ISMIR 2005 - 6th International Conference on Music Information Retrieval, pp288-295, 2005.

[5] Kathy Melih, Ruben Gonzales. Audio Retrieval Using Perceptually Based Structures. In Proceedings of the IEEE Conference on Protocols for Multimedia Systems and Multimedia Networking, PROMS-MmNet, pp338-347, 1998.

[6] K. Bosteels, E. Kerre. Fuzzy audio similarity measures based on spectrum histagrams and fluctuation patterns. In Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, pp361-365, 2007.

[7] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang. Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval. In Proceedings of IEEE International Conference on Multimedia and Expo, pp304-309, 2010.

[8] Frank Kurth, Meinard Muller. Efficient index-based audio matching. In IEEE Transactions on Audio, Speech and Language Processing, vol.16(2):382-395, 2008.

[9] Wei-Qiang Zhang, Jia Liu. Two stage method for specific audio retrieval. In Proceedings of ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4:IV85-IV88, 2007.

[10] Luke Barrington, Mehrdad Yazdani, Douglas Turnbull, Gert Lanckriet. Combining feature kernels for semantic music retrieval. In Proceedings of ISMIR 2008 - 9th International Conference on Music Information Retrieval, pp614-619, 2008.

[11] John Saunders. Real-time discrimination of broadcast speech/music. In Proceedings of ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2: 993-996, 1996.

[12] P. Pudil, J. Novovicova, J. Kittler. Floating search methods in feature selection. In Pattern Recognition Letters, vol. 15(11): 1119-1125, 1994.

[13] D. Hall, G. Ball. Isodata a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, 1965.

[14] L.A.R. Yamamoto, J.G. Beerends. Impact of network performance parameters on the end-to-end perceived speech quality. In Proceedings of EXPERT ATM Traffic Symposium, 1997.

[15] Marko Kos, Zdravko Kacic, Damjan Vlaj. Acoustic classification and segmentation using modified spectral roll-off and variance-based features. In Digital Signal Processing (DSP) 23(2):659-674, 2003.

[16] Xin Chen, Yunxin Zhao. Building Acoustic Model Ensembles by data sampling with enhanced trainings and features. In IEEE Transactions on Audio, Speech & Language Processing (TASLP) 21(3):498-507, 2013.

[17] Cyril Joder, Slim Essid, Gael Richard. A comparative study of tonal acoustic features for a symbolic level music-to-score alignment. In Proceedings of ICASSP 2010, pp409-412, 2010.

[18] Bo Shao, Mitsunori Ogihara, Dingding Wang, Tao Li. Music recommendation based on acoustic features and user access patterns. IEEE Transactions on Audio, Speech & Language Processing (TASLP) 17(8):1602-1611, 2009.

[19] Tao Li, Mitsunori Ogihara and George Tzanetakis. Music Data Mining. CRC Press, 2011.

[20] Tao Li, Mitsunori Ogihara, and Qi Li. A Comparative Study on Content-Based Music Genre Classification. In Proceedings of Annual ACM Conference on Research and Development in Information Retrieval (SIGIR 2003), Pages 282-289, 2003.