http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (1): 164-168, 2010
ISSN 1812-5638

A Novel Quick Audio Retrieval Method Based on Similarity

Jinglong Wu, Guanghui Ren and Peng Li
School of Electronics and Information Technology, Harbin Institute of Technology,
Harbin, Heilongjiang 150001, China

Abstract: In this study, we investigate to improve search speed and meanwhile guarantee robustness by a new method including 3 stages. Especially, in these stages, there is a kind of coarse retrieval method whereby the elements of features are tested to exclude mismatched points of a long audio signal at very high speed. Some tests have been implemented to compare with other methods with good performance. Experimental results demonstrate that the new method offers quicker retrieval speed and guarantees the robustness in additive noise environment.

Key words: Audio processing, audio retrieval, mel-frequency cepstral coefficients, coarse retrieval

INTRODUCTION

With the ever-increasing amount and availability of multimedia data, it is more and more important for us to find methods to detect and retrieval audio information from the mass data rapidly. At present, in practical applications the only way to find audio information is to search the label made by people manually, however this way using labels has several defects such as high error rate, time-consuming and impossibility of including all the information the audio files have. So we wish to find audio information through audio itself instead of labels. The method is named audio retrieval.

In audio retrieval area, there are two subfields which refer to as content-based retrieval and similarity-based retrieval. Most studies have dealt with content-based methods such as audio content classification (Kiranyaz *et al*., 2006), ASR (Automatic Speech Recognition) (Matthews *et al*., 2007). This kind of methods based on content extracts information with meaning instead of information in acoustic level. For examples, in audio content classification, audio usually is divided into speech, music and fuzzy; and in ASR, audio is translated to words of a language.

In contrast, our research interest is retrieval audio information based on similarity. More specifically, audio signal is divided into several segments and an example is used to find out the segments whose similarity with the example goes beyond a given threshold. In other words, we aim at signal matching rather than processing with signals' semantics.

The applications of the similarity-based retrieval include the detection and statistical analysis of broadcast music and commercial spots, the content identification, detection and copyright management of pirate copies of music clips.

In the studies which concern similarity-based retrieval (Lin *et al*., 2006), a conventional method based on histograms (Kashino *et al*., 1999) and a series of methods (Kashino *et al*., 2003; Kimura *et al*., 2008; Zhang and Liu, 2007; Kim *et al*., 2006) improved from it has been widely used. In these methods, adjacent histograms are extracted out from audio signal as features; they are strongly correlated with each other. So, unnecessary matching calculations are skipped by a similarity upper bound property of histograms. However, the search speed of the methods is very poor.

Therefore, we propose a new method with 3 stages to improve the search speed. In the first stage, audio classification technology is applied to reduce the searching time. The second stage is coarse retrieval which could remove numerous mismatching points using very little time and make the entire method obtain a very high speed. Finally, precise retrieval is used to guarantee the searching accuracy. The proposed method shows significantly faster searching speed when having sufficient accuracy.

Basic algorithm: The basic algorithm is the algorithm we improved from, which also refers to as direct audio search. Figure 1 shows diagram of the basic algorithm.

Figure 1 outlines the diagram of basic algorithm. In the algorithm, features are calculated from both audio signal and example firstly. In this figure, a column is a sequence of features extracted out of an audio signal frame and the number of rows is feature dimension. And

Corresponding Author: Jinglong Wu, School of Electronics and Information Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

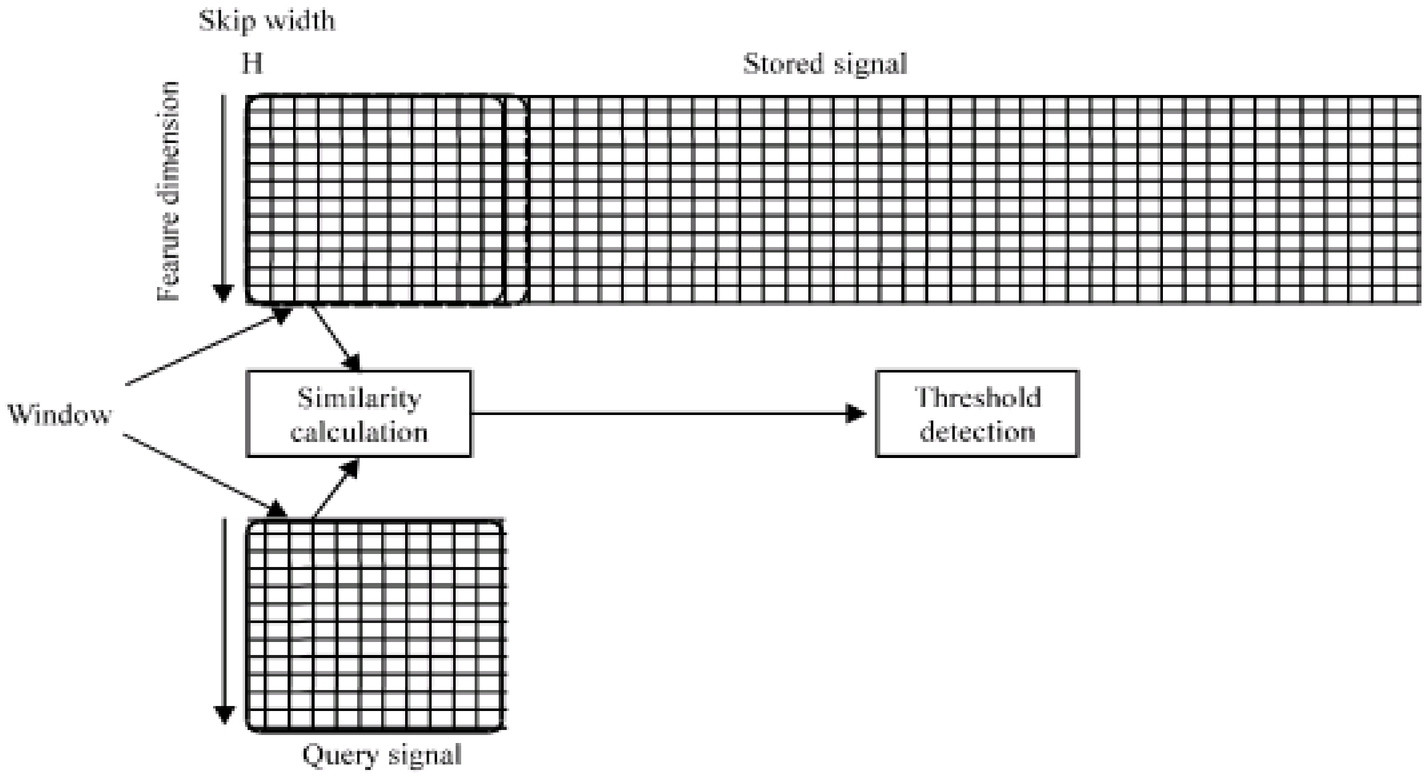

Fig. 1: Basic algorithm

then a series of given length windows (skip width is one) are applied to the features and the window is named feature frame which is different from the audio signal frame mentioned above. Through these windows, these features are transformed to many feature frames. Thirdly, similarity between the feature frame of example and which of the audio signal is calculated. If the similarity goes beyond a given threshold, it is thought that object is detected and located. Finally the window in the audio signal moves forward and the similarity calculation proceeds are repeated (Johnson and Woodland, 2000).

The improved search algorithm: Based on the basic algorithm, we propose a new algorithm to reduce the number of mismatched points step by step. After feature frames used in basic algorithm are obtained, the entire feature frames are classified into a certain number of categories firstly. In this way the mismatched categories could be got rid of by examining the category types merely in very short time. In the second part, it is also hoped to remove as many mismatched points as possible quickly. For this aim, a method called coarse retrieval is applied, which examines elements in the feature frames. In the final part, classical distance measure (Squared Euclidean distance) is used to ensure accuracy.

The improved search algorithm we introduced can be divided into following 6 steps:

Preprocessing stage:

(1) Extracting features from both audio signal and example.
(2) Applying windows to the features to obtain feature frames.

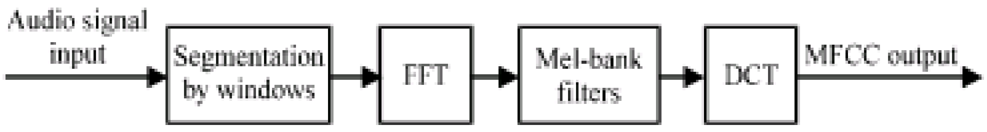

Fig. 2: Diagram of MFCC'

(3) Classifying the feature frames into a given number of types.

Retrieval stage:

(4) Comparing the type of example and which of audio signal, if the two types is the same, then search forward.
(5) Coarse retrieval.
(6) Precise retrieval.

Features: There are many types of features which are adopted in the audio retrieval area, such as zero-crossing rates of waveforms (ZCR), linear predictive coding coefficients (LPC), histograms based on short-time frequency spectrum and other multiple features (Kim *et al*., 2006). However, these features are not robust enough. Therefore, we use Mel-Frequency Cepstral Coefficients (MFCC) in our method. For being better understood, the MFCC is briefly introduced here.

In Fig. 2, firstly audio signals are segmented by windows such as hamming windows and rectangular windows. Secondly, we calculate frequency spectrum of the signals. Then the spectrum will pass a series of mel-bank filters. Finally, we use Discrete Cosine Transform (DCT) to reduce the coefficients' dimensions. As a result, we obtain the features which are named static MFCC (Zheng *et al*., 2001). The MFCC calculated from a window will compose a column in the feature matrix.

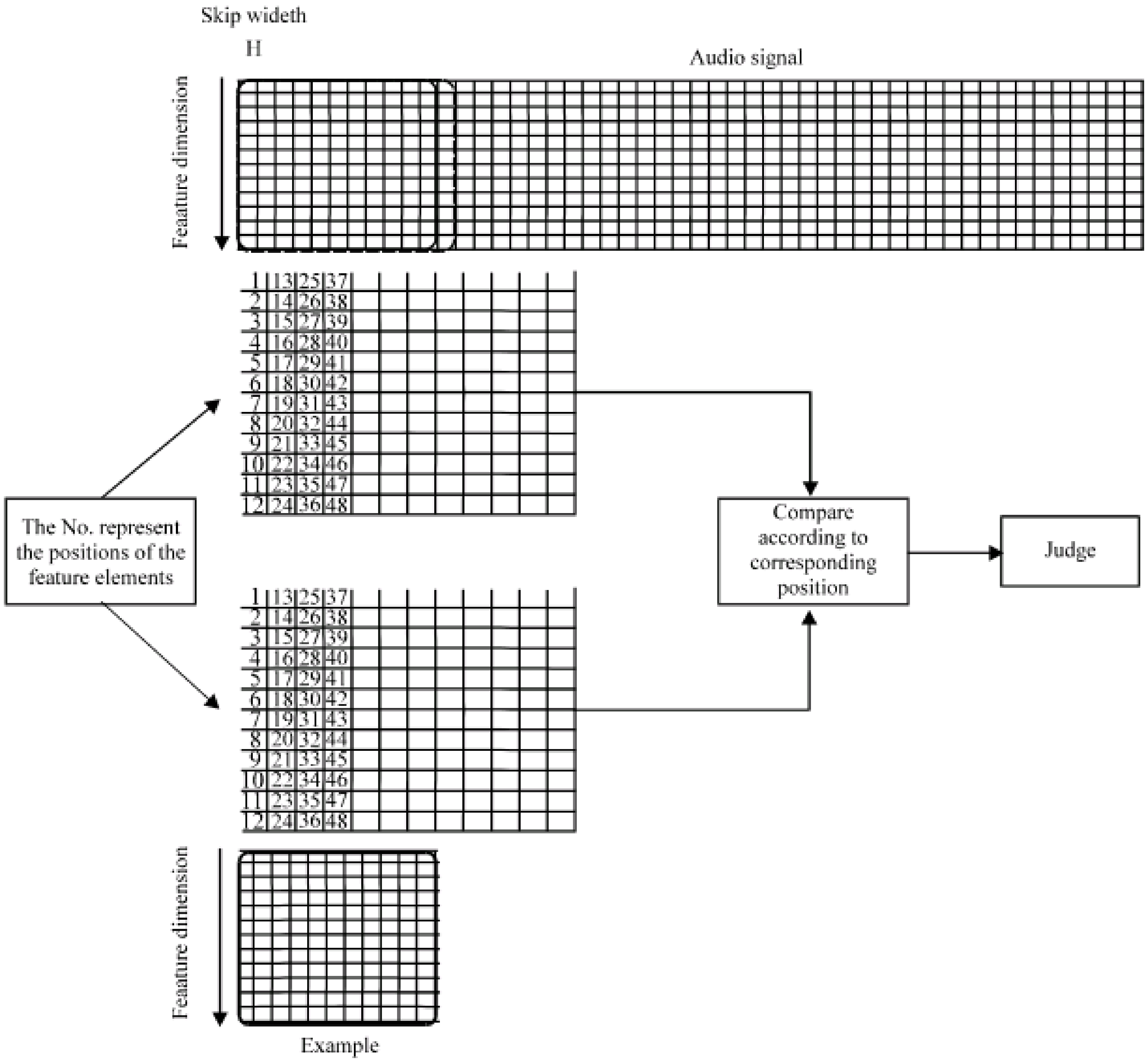

Fig. 3: Coarse retrieval

Classification: After the feature frames are obtained (concrete details have been given in the beginning of part III). They are classified into certain number of types, using Vector Quantization (VQ) technology. By this, we can just search the feature frames of the type which is the same with the type of example and neglect all mismatched points of other types. This process will reduce retrieval time, but it will also enlarge the false dismissal rate. Therefore we should choose the sum of types properly.

The procedure we introduced above is called preprocessing stage.

Coarse retrieval: In this process, the feature frames dealt with are the frames whose type is the same as example. In order to search rapidly, the distance between frame of audio signal and which of example is not computed directly. We compare the feature's element of corresponding position firstly. Because it is noticed that not only are features robust, but the elements of features are robust as well.

Figure 3 outlines the coarse retrieval, which highly accelerates speed of the proposed method. In the Fig. 3, the numbers represent elements' positions in the feature frame. Firstly, elements marked number 1 are compared. If difference between the two elements does not exceed a given threshold, number 2 elements are compared successively until the end element. Otherwise, if the difference exceeds the threshold, the feature frame which has the element is not the object be found, then the feature frame is skipped and the next goes forward.

However, comparing all the elements is time-consuming and will increase the false dismissal rate. Moreover, if the number of elements be chosen is too low, more feature frames would be preserved and more time would be spent in precise retrieval. The number of elements should be compromised with false dismissal rate and search speed. The concrete value and threshold will be given in experiment part.

Precise retrieval: The last step of retrieval is the step which we decide the feature frame is object or not. In this step, Squared Euclidean distance is used as the similarity measurement. When the distance exceeds a given threshold, the feature frame is the object:

$$\text{Distance} = \sum_{n=1}^{N}(x_{1n} - x_{2n})^2 \qquad (1)$$

where, N is the number of elements in one feature frame, x_{1n} and x_{2n} is the element in the audio signal and example, respectively.

EXPERIMENTS

Two types of experiments are conducted to evaluate performance of the proposed method using a Laptop (Pentium M 1.6 GHz, Windows XP). First is evaluating search speed, the second is search accuracy.

Search speed: In this experiment, audio signal is a broadcast recording of 8.5 h; examples are 8 commercial advertisements chosen randomly from another recording file. All examples are around 12.5 sec. Each example repeats 4 to 16 times in the audio signal. File format is wav and sampling rate is 8 kHz.

When we experiment about search speed, preprocessing is distinguished from retrievaling. The preprocessing stage is performed before example is given. The more detail information has been introduced in part II B. When we refer to search speed, it specifically means the time used in step 4, 5 and 6.

The time consumed in preprocessing stage is also very important. In our experiment, extracting feature from 1 h audio file needs 244 sec, classification needs 0.8 sec. It means preprocessing stage needs 6.8% of audio length.

Then some specific experiment data, which are related to search speed, used in our experiment will be given. In the stage 1, frame length is 2.5 sec, frame skip is 0.25 sec. A codebook with 32 code words has been generated by VQ (LBG algorithm), it means feature frames are classified into 32 categories; and the classification is according to the first 3 columns elements of feature frames. When coarse retrieval, we only use first 4 columns elements and coarse retrieval threshold is set to 4.

Figure 4 is the snippet of the entire experiment; it's about 15 min retrieval result. The information of entire experiment will be given as data other than figure. In the figure, the points marked with circles are reserved points after using category type comparison and coarse retrieval. The other points are the result of exhaustive search. It is noticed that most of mismatched points are got rid of after comparing category type and coarse retrieval method. In the entire experiment, the possible points (still possible to be matched) are reduced to approximately 1/500 in comparison with exhaustive search. The CPU time is also shortened by 1/190. It shows that the proposed method takes only 4 msec for searching 8.5 h audio signal.

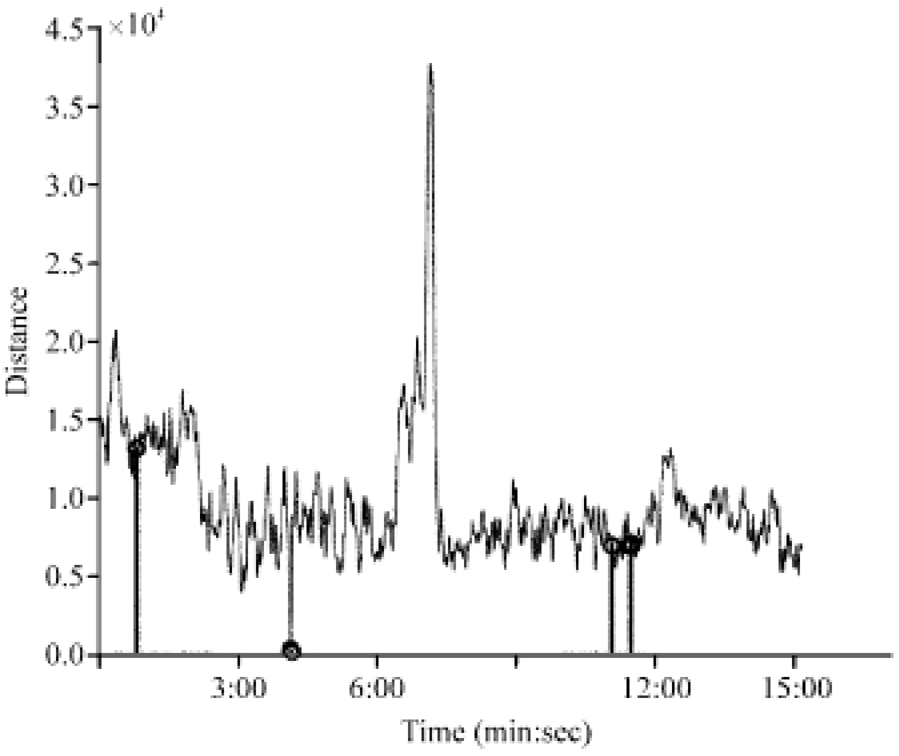

Fig. 4: The result of proposed search algorithm

Table 1: Comparison on search speed

Composition	The proposed method Pentium M	Conventional time-series active search method	Method in reference (Kimura *et al*., 2008) Pentium IV
Experimental condition	1.6 GHz	SGI O2	2.0 GHz
Search speed (m sec^{-1} h)	0.47	333	1.5

Several experiments are done for the comparison on search speed with other two audio retrieval methods. The first method is the conventional Time-Series Active Search (named TAS) method which is based on histogram. The second is The Method Based on a Piecewise Linear Representation of Feature Trajectories which is a method improved from TAS and which is also the fastest method have been published. The result of comparison experiments shows that the proposed method improves the search speed significantly as compare to the two other methods in the experiments.

The result of comparison experiments is shown in Table 1. The data of two other methods is quoted from their papers (Kashino *et al*., 1999; Kimura *et al*., 2008) directly. For better comparison, experimental condition is given out.

Accuracy: In the experiment testing accuracy (Correct ratio), the material we used is the same with which of the search speed test. For testing robustness of the method, different white gauss noises have been added in all the examples. The test result is showed in Fig. 5 where correct ratio is the average of losing ratio and fallout ratio and horizontal axis is signal-to-noise ratio. The result of accuracy experiment demonstrates that correct ratio could reach 95% when SNR is more than 15 db.

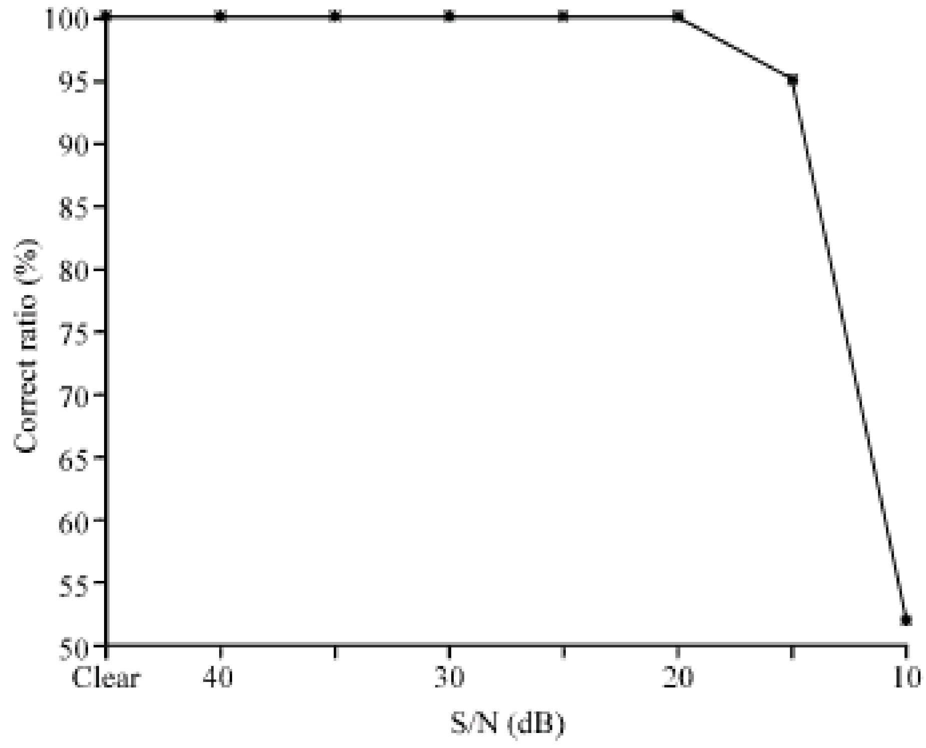

Fig. 5: Search accuracy

$$\text{Correct ratio} = (\text{losing ratio} + \text{fallout ratio})/2 \quad (2)$$

CONCLUSIONS

This study proposes a search method that can quickly detect and locate known examples in a long audio signal. The method enhances the search speed performance by several improvements.

Firstly, we discuss about feature selection. Previously methods adopt single feature such as ZCR and short-time frequency spectrum often were used for simple calculation. In our method, it is noticed that the robustness is more important than simple calculation for better performance. Therefore, the proposed method used mel frequency cepstrum coefficient as audio feature. By using more robust feature, we can enlarge the frame and skip width for quicker searching.

Next, we deal with the speed performance derived from the category type comparison. Because we can gain the category type in the preprocessing stage previously, it can be used shorter time for making a preliminary decision in the processing stage. However, it can also infect the reduction of search accuracy if sum of types is too large.

Finally, we deal with the speed performance derived from the coarse retrieval. We advance the method for the fact that the elements of feature also have robustness. They do not vary much by adding small noise. Therefore the difference of elements is calculated to ensure they do not vary too much or exceed a given threshold. This method remarkably improves the search speed compared with other two methods. However, when the number of elements used in the coarse retrieval is relatively small, the feature frames can not be distinguished very well. On the other hand, when the number is relatively large, the proposed method need more time in coarse retrieval and it can not greatly improve the search speed compared with other two methods.

As a result of tests, the search speed is approximately 3 times quicker than time-series algorithm meanwhile to keep robust. In the future of the study, we wish to find more robust feature and broaden the domain of application of audio retrieval.

REFERENCES

Johnson, S.E. and P.C. Woodland, 2000. A method for direct audio search with applications to indexing and retrieval. Proceedings of the Acoustics, Speech and Signal Processing, Jun. 5-9, IEEE, Istanbul, pp: 1427-1430.

Kashino, K., G. Smith and H. Murase, 1999. Time-series active search for quick retrieval of audio and video. Proc. ICASSP, 6: 2993-2996.

Kashino, K., T. Kurozumi and H. Murase, 2003. A quick search method for audio and video signals based on histogram pruning. IEEE Trans. Multimedia, 5: 348-357.

Kim, K.M., S.Y. Kim, J.K. Jeon and K.S. Park, 2006. Quick audio retrieval using multiple feature vectors. IEEE Trans. Consumer Electr., 52: 200-205.

Kimura, A., K. Kashino, T. Kurozumi and H. Murase, 2008. A quick search method for audio signals based on a piecewise linear representation of feature trajectories. IEEE Trans. Audio Speech Language Proces., 16: 396-407.

Kiranyaz, S., A.F. Qureshi and M. Gabbouj, 2006. A generic audio classification and segmentation approach for multimedia indexing and retrieval. IEEE Trans. Speech Audio Proces., 14: 1062-1081.

Lin H., Z.J. Ou and X. Xiao, 2006. Generalized time-series active search with kullback-leibler distance for audio fingerprinting. IEEE Signal Proces. Lett., 13: 465-468.

Matthews, B., U. Chaudhari and B. Ramabhadran, 2007. Fast audio search using vector space modeling. Proceedings of the IEEE Workshop on Speech Recognition and Understanding, ASRU, Dec. 9-13, IEEE Kyoto, pp: 641-646.

Zhang, W.Q. and J. Liu, 2007. Two-stage method for specific audio retrieval. Proceedings of the Acoustics, Speech and Signal Processing, Apr. 15-20, Honolulu, IEEE, pp: IV-85-IV-88.

Zheng, F., G.L. Zhang and Z.J. Song, 2001. Comparison of different implementations of MFCC. J. Comput. Sci. Technol., 16: 582-589.