Journal of Computer Science 2 (3): 297-302, 2006
ISSN 1549-3636

Real-Time Audio Retrieval Method and Automatic Commercial Detecting System

[1,2]Guibin Zheng and [1]Jiqing Han
[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]School of Computer Science and Technology, Harbin University of Science and Technology, China

Abstract: This study presents a robust audio retrieval method and discusses its control strategy. In the method, the retrieval target is divided into short segments, each segment is searched respectively and a retrieval window is used to maintain a list of segments that can be searched simultaneously. The method can quickly detect and locate known sound in real-time audio stream, multimedia archives or the Internet. It can maintain high performance even if large part of target is absent in the input stream. Its retrieval speed can be adjusted by the length of retrieval window and is independent on target length. Using the proposed method, an automatic commercial detecting system is developed, which is used to detect and locate advertisements from real-time TV or radio broadcast audio signal and estimate broadcast length of them. The recall rate and precision rate of the system are 100 and 99.7% respectively.

Key words: Audio retrieval, commercial detecting, precision rate, recall rate

INTRODUCTION

Rapid development of multimedia and network technology has resulted in large and ever-increasing stores of multimedia data. Therefore efficient means to index and retrieve multimedia data is required in order to fully use multimedia information. With the increase in quantity of multimedia data, this technology will be needed more and more urgently. In recent years, audio, as an important media, has been paid more and more attentions.

It is a kind of useful audio retrieval to search quickly through a long audio or video stream (termed an input stream) to detect and locate a known target audio signal (termed a reference template). There are several applications where it is required. One application is monitoring occurrences of a commercial or other specified audio segment from real-time TV or radio signal. Another is searching and retrieval of music or other interested audio fragment from unlabeled multimedia archives or the Internet[1,2].

Even if a reference signal is known, the detection is still difficult when only part of reference template presents in the input stream or when the noise cannot be ignored. Whereas users often need the ability to detect the occurrence of partial reference template from the input stream and estimate the length of it. For instance, advertisers are rather concerned about whether their commercials have been broadcast according to the contract, including the broadcast length, frequency and time.

Smith et al. proposed an audio retrieval method[1,2], which used an active search algorithm and histogram modeling of zero-crossing features[1] or power spectrum[2]. Johnson and Woodland[3] also developed a fast retrieval method. Since the reference template is searched as a whole unit, those methods cannot detect the occurrence of partial reference template. Furthermore those methods are not accurate enough under practice circumstance[1] and performance decreased dramatically when SNR (Signal-To-Noise Ratio) is less than 30dB[2] or even 40dB[1], while 30dB should be a rather better SNR level in practical circumstance. And the robustness to noise was not discussed in Johnson and Woodland[3].

Segmentation based retrieval method: In order to detect the target when only a fragment of it presents in the input stream, estimate the length of found part of target easily, simplify the search complexity and improve robustness to noise, a long reference template is divided into a series of little segments and each segment is searched respectively, the process is shown in Fig. 1. According to the presence of segments, the occurrence of original reference template can be decided. Segmentation in our method is different from that for audio content analysis described[4-7] which divides audio data into segments on the border of different audio classes such as speech, music and so on and labels each segment a certain audio class. In our method, segmentation and control of retrieval window will impact the retrieval performance directly.

Segmentation: In segmentation, the length of each segment can be different but head of the first segment and tail of the last segment must coincide with the head or tail of reference template. Given the number of segments is N, segments are numbered from 1 to N in sequence. Let $len(i)$ represents the length of segment i and $d(i,j)$ denotes segment interval between segment i and j, then

Corresponding Author: Guibin Zheng, School of Computer Science and Technology, Harbin Institute of Technology, China

$$d(i,j) = \sum_{k=i}^{j-1} d(k,k+1) \quad i \in [1,N-1], \quad j \in [i+1,N], \quad d(i,i) = 0 \tag{1}$$

In the most simplest case, all segments are divided at equal time interval with the same segment length, i.e. $len(i) = L$, $d(j\text{-}1,j) = D$ (i = 1, 2, …N, j = 2,3,…N), and $d(i,j) = |j\text{-}i|D$, where L and D are constant. The time granularity of retrieval method is decided by the segment interval D. In the process of segmentation, similarity threshold of each segment is also calculated with the silent frame ratio of the segment is considered,

$$th[i] = TH + \max(0, sfr[i] - \beta)(1 - TH) \tag{2}$$

where $th[i]$ is the threshold of segment i, TH is a constant corresponding to the threshold of segments without silent frame, $sfr[i]$ is silent frame ratio of segment i and β is a constant.

Retrieval window: Since segments of reference template are sequential, segment with little sequence number should presents prior to segments with greater sequence number. Therefore a retrieval window is used to maintain a list and control the number of segments that can be searched simultaneously. Given the length of retrieval window is M, i.e. there are M segments can be searched simultaneously. Before retrieval, initialize retrieval window firstly: register sequence number of segment from 1 to M orderly. During retrieval, when there is a segment has been spotted, retrieval window will be adjusted according to the spotted segments and the time elapsed, so that the retrieval window can be time synchronous with the input stream, which is important for normal retrieval of reference template.

Referential sequence number is used to decide which segment can be registered into retrieval window. Assuming segment i is the last spotted segment at time t_d (i) and current time is t_{cur}, referential sequence number k can be calculated as follow:

$$k = \max(k', 1)$$

$$k' = \max\{ j \mid d(i+1, j) \le t_{cur} - t_d(i), j = i+1, \ldots, N \} - \lfloor \lambda \cdot M \rfloor \tag{3}$$

where λ is a coefficient considering that some segments may be lost in the input stream and the possible misadjustment of retrieval window incurred by wrong detection of segment. From segment k, M unspotted segments will be registered into the retrieval window. It is a simple case shown in Fig. 1 (assuming all segments after segment k have not been spotted).

Only when at least one segment has been spotted in the searching process, can the retrieval window be adjusted so that the rear segments can be registered into the retrieval window and have chance to be spotted. Then it must be ensured that at least one of segments registered in the retrieval window can be spotted, otherwise all segments will not be spotted and the occurrence of reference template will not be found too. Therefore, the retrieval window should be long enough. Assuming the segment spotting is probability event, its probability, i.e. segment recall rate, varies with

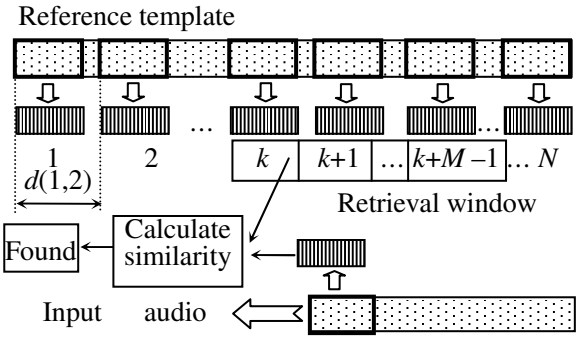


Fig. 1: Scheme of segmentation based retrieval method

different channel quality (noise level) and segment spotting method. If the recall rate of segment is Rr_{seg}, then the probability of that at least one segment in retrieval window can be detected, Hr_{win} (Hit rate of retrieval window) is,

$$Hr_{win} = 1 - (1 - Rr_{seg})^M \tag{4}$$

If restricting the minimum value of Hr_{win} to $Hr_{win}^{\min}$, the length of retrieval window can be computed using the following formula according to above equation,

$$M \ge \lfloor S_f \cdot \ln(1 - Hr_{win}^{\min}) / \ln(1 - Rr_{seg}) \rfloor \tag{5}$$

where S_f is safety factor, $S_f > 1/(1-\lambda)$. Bigger retrieval window length will favor the recall rate of retrieval but will also slow down the retrieval speed because more segments should be search at the same time. So the length of retrieval window should be a tradeoff between recall rate and retrieval speed.

False detection of segment will make the retrieval window be adjusted by mistake; consequently, the succeeding occurrence of reference template may not be found. Although the problem has been considered in equation (3) by introducing a coefficient λ, but it is not enough to solve the problem. Using following rules in addition can solve the problem:

Rule 1: Calculate equation (3) using each spotted segment and use the average value as the referential sequence number.

Rule 2: Initialize the retrieval window if a required number of segments cannot be found in limited time length.

Because the false detection rate of segment is generally low, the average value of equation (3) can be more reliable so that the retrieval window can be adjusted more correctly using rule 1. Since the recall rate of segment is usually high, the number of spotted segments within fixed time length will be greater when the reference template presents. Rule 2 can cancel the wrong adjustment of retrieval window as early as possible.

Retrieval of reference template: When one segment has been spotted, the retrieval window will slide on reference template from the front to the end synchronously with the input stream. If the reference template does present in the input stream, the last segment should present within the time length of the reference template and retrieval conclusion of whether find reference template or not can be drawn at that time. It is a sign of that reference template has ended in most cases if no segment is found in a relative long time, so the retrieval conclusion should also be drawn then. It can be concluded that the reference template has occurred if the number of found segments is greater than a threshold. The threshold is a fixed constant or proportional to the total number of segments.

The length of reference template found in the input stream can be evaluated easily by calculating the time span between found time of the first and the last spotted segments. Even if a fragment of reference template occurs in the input stream, the retrieval method can also detect it normally if the number of found segments is greater than the threshold.

CONTROLS IN REAL-TIME RETRIEVAL

Control of retrieval response time Lag (RRTL): In real-time audio retrieval, when the reference template is found in the input stream, some action would be triggered, such as recording the TV or radio broadcast signal containing the reference template into multimedia files for late verification in commercial detecting system. We call those actions retrieval responses. But when the occurrence of reference template is reported, the target signal has flowed away, so a buffer is in need to store the signal beforehand. And the length of buffer must be longer than that of the longest reference template. In commercial detecting system, the longest commercials we have found are about 10 minutes long and most of them are in some radio stations and local TV stations where the cost of commercial is relative low. In this case, the buffer size of an 8-channel TV commercial detecting system should be about 400-500M Bytes at a medium compression ratio with less CPU load or about 200M Bytes at a high compression ratio (e.g. real media format, .rm) with heavy CPU load.

The time length from the start time of reference template to the time when retrieval response starts up is called retrieval response time lag (RRTL), depicted in Fig. 2. If RRTL can be shortened, the buffer size will be reduced effectively. To shorten RRTL is a very difficult problem for retrieval methods where the reference template is searched as a whole unit. The problem can be solved in our method using a proper control strategy.

Startup of retrieval response: False detection of segment usually takes place when the reference template does not present in the input stream. And the

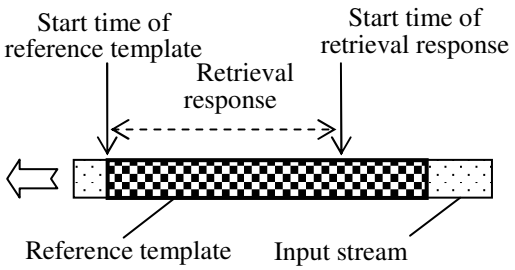


Fig. 2: Retrieval response time lag

time interval between two false detections is often much more longer than segment interval and segment length. Thus it can be a positive indication of reference template occurrence when a certain amount of segments are spotted within a limited time length. Our solution for reducing the RRTL is: start up the retrieval response when *m* segments have been spotted from *n* continuous segments. Considering that the retrieval response may be triggered by mistake, following rule is introduced:

Rule 3: Cancel the retrieval response if reference template cannot be found in time length of reference template.

Estimation and control of RRTL: To estimate the value of RRTL, *n* continuous segments of reference template are treated as a block (depicted in Fig. 3). If the first segment in a block, e.g. block *i*, is spotted and at least *m*-1 following segments in the block can be spotted, the retrieval response is regarded to be triggered by that block. Each block has the same ability to activate retrieval response. Corresponding probability is:

$$P(R \mid B_i) = Rr_{seg} \sum_{j=m-1}^{n-1} C_{n-1}^{j} Rr_{seg}{}^{j} (1-Rr_{seg})^{n-1-j} = P_{block}(R), \quad i \in [1,\ N-n+1] \tag{6}$$

where R denotes the event that retrieval response is triggered, B_i denotes block *i*. The value of $P(R \mid B_i)$ is same for different block, which is a constant $P_{block}(R)$.

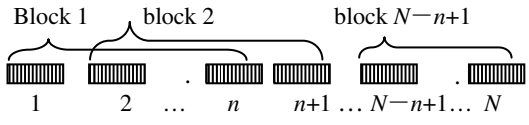


Fig. 3: Plotting the reference template into blocks

Only if blocks from 1 to *i*-1 have not triggered the retrieval response, can block *i* has chance to trigger the retrieval response. Corresponding probability is:

$$P(RB_i) = P(R \mid B_i)P(B_i) = P_{block}(R) \cdot \prod_{j=1}^{i-1} (1 - P(RB_j)), \quad i \in [2,\ N-n+1] \tag{7}$$

where $P(B_i)$ is probability that block i has the chance to trigger the retrieval response, therefore $P(RB_1) = P(R \mid B_1) = P_{block}(R)$ and value of $P(RB_i)$ can be computed recursively. The probability of that retrieval response can be triggered is:

$$P(R) = \sum_{i=1}^{N-n+1} P(RB_i) \tag{8}$$

If the retrieval response has been triggered by block i, the probability is:

$$P(B_i \mid R) = \frac{P(RB_i)}{P(R)} = P(RB_i) \Big/ \sum_{i=1}^{N-n+1} P(RB_i) \tag{9}$$

The expectation of block sequence number that triggers retrieval response can be computed as follow:

$$\begin{aligned} E_{block} &= \sum_{i=1}^{N-n+1} i \cdot P(B_i \mid R) \\ &= \sum_{i=1}^{N-n+1} \left[i \cdot P(RB_i) \Big/ \sum_{i=1}^{N-n+1} P(RB_i) \right] \end{aligned} \tag{10}$$

If a block triggers retrieval response, the probability of that the ith segment of the block triggers retrieval response, i.e. the segment is the mth spotted segment in the block, is:

$$\begin{aligned} P(RS_i) &= Rr_{seg}{}^2 \cdot C_{i-2}^{i-m} Rr_{seg}{}^{m-2} (1 - Rr_{seg})^{i-m} \cdot \\ &= C_{i-2}^{i-m} (1 - Rr_{seg})^{i-m} \cdot Rr_{seg}{}^m, \quad i \geq m \end{aligned} \tag{11}$$

where S_i represents segment i in the block. The expectation of segment sequence number that triggers retrieval response in the block can be computed as follow:

$$\begin{aligned} E_{seg}^{blcok} &= \sum_{i=m}^{n} i \cdot P(RS_i) \Big/ P_{block}(R) \\ &= \sum_{i=m}^{n} [i \cdot C_{i-2}^{i-m} \cdot Rr_{seg}{}^m (1 - Rr_{seg})^{i-m} \Big/ P_{block}(R)] \end{aligned} \tag{12}$$

The expectation of segment number that triggers retrieval response in reference template can be computed as follow:

$$E_{seg} = \lfloor E_{block} + E_{seg}^{blcok} - 1 \rfloor \tag{13}$$

The expectation of RRTL is:

$$RRTL = d(1, E_{seg}) + len(E_{seg}) \tag{14}$$

In the most simplest case described in section 1.1 where $d(i,j) = |j-i|D$ and $len(i) = L$, above formula can be simplified as follow:

$$RRTL = (E_{seg} - 1)D + L \tag{15}$$

A suitable strategy can be made for practical application according to equation (15) to limit RRTL to an acceptable level by selecting proper m, n and the segment interval.

Capability control of real-time retrieval system: Another important issue in real-time retrieval system is that the retrieval speed must be not slower than that of the input stream. Two aspects of the problem should be solved: (1) confirm the maximum number of reference templates the system can search simultaneously, $N_{Ref}^{\max}$, so as to search as many targets as possible; (2) given the number of reference templates to search, confirm the maximum value of retrieval window, $M_{\max}$, in order to obtain the best recall rate and precision rate. Both of the above aspects are about making the best use of system computing resource under a reliable load limit.

In segmentation based retrieval method, it is easy to test how many segments system can search simultaneously in real time. Let the value is $N_{seg}^{\max}$, then $N_{Ref}^{\max}$ can be computed as follow:

$$N_{Ref}^{\max} = \lfloor N_{seg}^{\max} / M \rfloor \tag{16}$$

where M is acceptable retrieval window length, at which length the system can obtain satisfying performance (recall rate and precision rate). $M_{\max}$ can be obtained as follow:

$$M_{\max} = \lfloor N_{seg}^{\max} / N_{Ref} \rfloor \tag{17}$$

Automatic Commercial Detecting System (ACDS)

A commercial detecting system (ACDS) has been developed based on the proposed method and the system can be run in real time using typical personal computers. The ACDS can detect appointed advertisements from 8-channel real-time TV or radio broadcasting and record the broadcast signal into multimedia files that is stored in a database together with detecting results, such as time, length, station name and so on. The architecture and system interface are shown in Fig. 4 and 5.

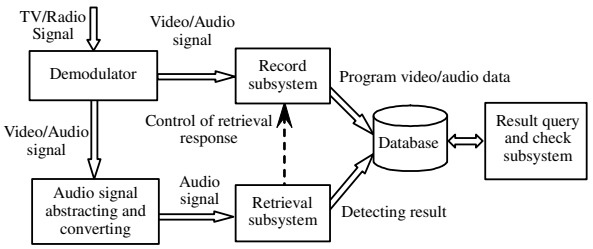


Fig. 4: Architecture of automatic commercial detecting system

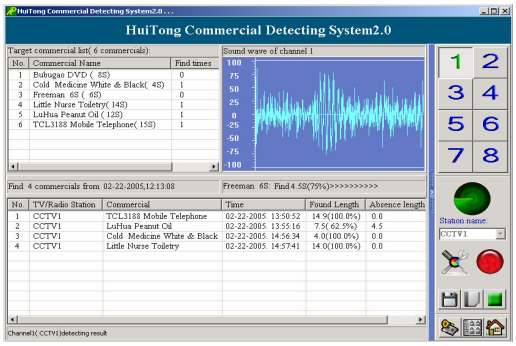


Fig. 5: Main interface of automatic commercial detecting system

PERFORMANCE EVALUATIONS of ACDS

To evaluate the proposed approach in real-time audio retrieval, experiments are performed and shown via ACDS. Robust feature MFCC (Mel Frequency Cepstrum Coefficient) and cosine distance are adopted in the system. For simplicity, reference template is divided evenly into one-second segments that can be detected at a high recall rate and precision rate. Segment recall rate and precision rate obtained by experiment are 95.2 and 75% respectively under the practical circumstance. Considering that time granularity of 0.5 sec is enough for commercial detecting, segment interval of 0.5-sec is used.

In experiments, multimedia files recorded from CCTV1 advertisement program, 15 hours long in total, are used as input stream.

Retrieval speed and RRTL evaluation: The experiment is performed to test the retrieval speed under different retrieval window length (RWL) and different reference template length. Retrieval time comprises (1) reading data from disk time; (2) feature extraction time and (3) search time. Retrieval speed is expressed by the ratio of the multimedia file length (15 hours) to the retrieval time, i.e. times of real-time speed (xRT). Retrieval speed is tested on an ordinary personal computer (Pentium IV 2.4GHz CPU). For comparison, the method that searches the whole target(reference template) directly (SWTD)[8] is also tested. Experimental results are shown in Fig. 6. In our method, retrieval speed is only dependent on RWL. Therefore, the retrieval speed can be easily controlled by RWL no matter the length of reference template. When RWL is 5, the system can detect 275 different advertisements simultaneously in real time. With the increasing in reference template length, the retrieval speed of SWTD slows down quickly.

In the experiment on RRTL evaluation, specified advertisements are detected and located from real-time play of the multimedia files and retrieval response is triggered when two (m=2) segments have been spotted from five (n=5) continuous segments. The average RRTL is 1.78 sec and the false rate of startup of retrieval response is 1.3 and 95% erroneous startups are cancelled within 4.5 sec.

Retrieval recall rate and precision rate evaluation: The recall rate is defined as the number of correctly retrieved objects divided by the number of objects that should be retrieved. The precision rate is defined as the number of correctly retrieved objects divided by the number of all retrieved objects. Length of retrieval window affects not only retrieval speed, but also the recall rate and precision rate. The performance results of ACDS in the practice are shown in Fig. 7. When retrieval window length is 5, the recall rate and precision rate are 100 and 99.7% respectively, which is much higher than that of[8].

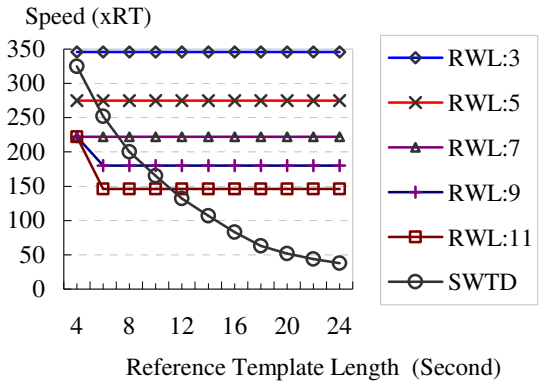


Fig. 6: Retrieval speed under different retrieval window length (RWL) and reference template length

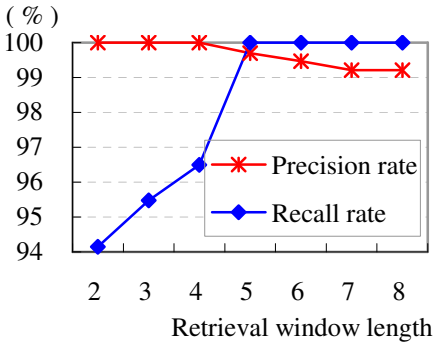
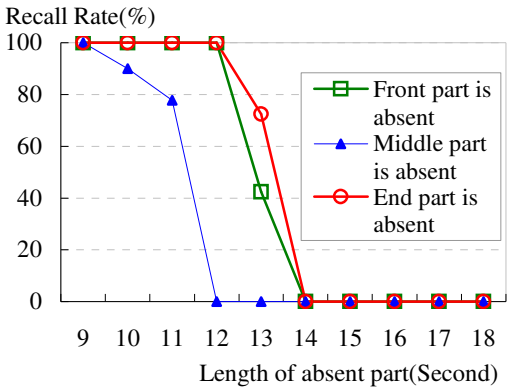


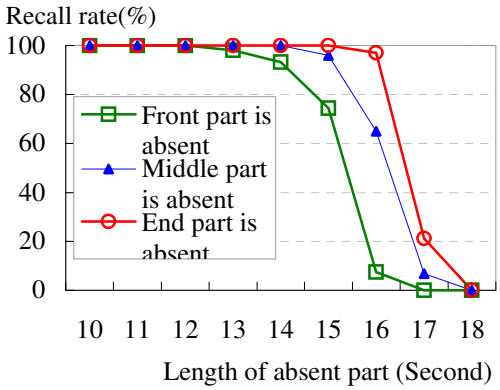
Fig. 7: Retrieval performance at different retrieval window length

Robustness to absence of partial reference template: Ten 20-sec commercials are used as reference templates to evaluate the robustness to absence of partial reference template when the front part, middle part or end part of reference template are absent. All reference templates contained in the multimedia files are complete. In order to imitate the absence of partial reference template, a segment of music that does not occur in the multimedia files is used to replace the front, middle or end part of those commercials to construct new ones and use them as reference templates.

In the experiment, the retrieval window is divided into two parts: size of the first part is 5 and the second part 7. Segment sequence numbers registered in the first part of retrieval window is continuous (e.g. 1,2,3,4,5) and those registered in the second part have an increasing interval (e.g. 8,12,16,21,26,31,36). If the number of found segments is greater than 4 (2.5 sec long at least, about one-eighth of reference template length), conclusion of finding reference template can be drawn. Experimental results are shown in Fig. 8. Even if the large part of reference template is absent, the method can also detect it. However, the methods described[1-3] would fail in this case.



a. Searches the whole target directly (SWTD)[8]



b. Segmentation based retrieval

Fig. 8: Robustness to the absence of partial reference template. The length of all reference templates is 20 sec and the average precision rate under all the three conditions in both methods is 99.2%

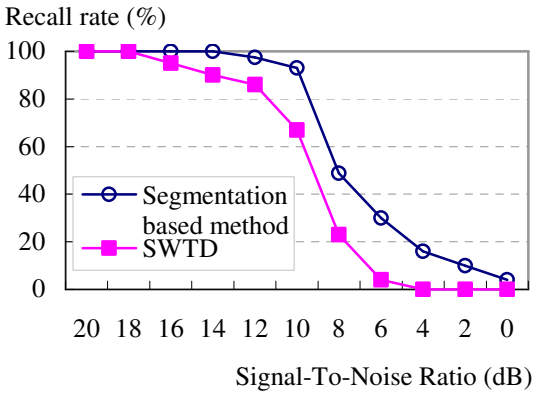


Fig. 9: Robustness to noise. Both of the two methods use MFCC feature and cosine distance. The average precision rates of the two methods under different SNR level are 91.4 and 89.5%, respectively

In the method we proposed, retrieval window length, control strategy and segment number threshold can be adjusted easily to satisfy different requirement for retrieval.

Robustness to noise: Robustness to noise has also been evaluated and experimental results are shown in Fig. 9. The performance of the method we proposed is best compared with that of the searching reference template as a whole unit and that of [1,2] whose accuracy deteriorates rapidly when SNR is less than 30dB.

With the decreasing in SNR, the recall rate of segment also declines. But if enough segments can be spotted, then the reference template can still be retrieved. Therefore, the recall rate of the proposed method is higher.

CONCLUSION

This study proposes a robust audio retrieval method based on segmentation and analyzed the control strategy in real-time retrieval. The method can detect and locate a known reference audio quickly from real-time audio stream or multimedia files and can estimate easily the length of retrieved result. The distinct features of the method are (1) even if large part of target audio is absent in the input stream, the method can also maintain a high recall rate and precision rate, which problem has not been discussed in the past; (2) RRTL can be reduced effectively, e.g. reduced to an average of 1.78 sec in our experiment; (3) the retrieval speed can be adjusted by the length of retrieval window; (4) the method is robust to noise. A commercial detecting system has been developed using the method, which works well at real-time TV and radio commercial detecting.

REFERENCES

1. Gavin, S., H. Murase and K. Kashino, 1998. Quick audio retrieval using active search. Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing. 6: 3777-3780.
2. Kashino, K., G. Smith and H. Murase, 1999. Time-series active search for quick retrieval of audio and video. Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing. 6: 2993-2996.
3. Johnson, S.E. and P.C. Woodland, 2000. A method for direct audio search with applications to indexing and retrieval. Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing. 3: 1427-1430.
4. Zhang, Y. and J. Zhou, 2004. Audio segmentation based on multi-scale audio classification. Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing. 4: 349-352.
5. Zhang, T. and C.-C.J. Kuo, 2001. Audio content analysis for online audiovisual data segmentation and classification. IEEE Trans. on Speech and Audio Processing. 9: 441-457.
6. Lu, G. and T. Hankinson, 2000. An investigation of automatic audio classification and segmentation. Proc. of 5th Intl. Conf. on Signal Processing. 2: 776-781.
7. Lu, L., H.-J. Zhang and H. Jiang, 2002. Content analysis for audio classification and segmentation. IEEE Trans. on Speech and Audio Processing. 10: 504-516.
8. Spevak, C. and E. Favreau, 2002. SOUNDSPOTTER— A prototype system for content-based audio retrieval. Proc. of the 5th Intl. Conf. on Digital Audio Effects (DAFx-02). Hamburg, Germany, pp: 27-32.