Highlighted Technical Paper C

Spoken Sentence Retrieval Based on MPEG-7 Low-Level Descriptors and Two Level Matching Approach

Jhing-Fa Wang Po-Chuan Lin Jun-Jin Huang Li- Chang Wen

wangjf@csie.ncku.edu.tw tony178@ms24.hinet.net kjtom@icwang.ee.ncku.edu.tw lcw@icwang.ee.ncku.edu.tw

Department Of Electrical Engineering, National Cheng Kung University.

701 No.1, Ta-Hsueh Road, Tainan City Taiwan R.O.C.

Tel: 886-6-2746867, Fax: 886-6-2387092

ABSTRACT

In this paper, we propose a spoken sentence retrieval system based on MPEG-7 audio LLDs (Low-Level Descriptors). Our system retrieves the spoken sentence by a two-steps sentence matching method. First, we locate several possible segments that are similar with the user's query in spoken documents and retrieve top N from these candidate segments. Secondly, we rank the top N candidates with rank-based method. Without using the large-vocabulary recognizer, the computational power can be greatly reduced and be more suitable for hand-held devices by using our approach. Furthermore, the use of MPEG-7 based features has been proven comparable with the MFCCs. (Mel-Frequency Cepstrum Coefficients) in the experiment results.

Keywords

Speech retrieval, Matching algorithm, MPEG-7, Audio features

1. INTRODUCTION

With the development of electrical technology and Internet, we have lots of opportunities to access various kinds of multimedia data. Among them, spoken data is one of the most widespread information, such as broadcast radio, television programs, video tapes, digital libraries, course, and voice recordings and so on. Although parts of them have been digitalized, it still lacks efficient ways to retrieval them. About these demands, information retrieval technologies provide users a lot of methods [7].

Also, with the advances in speech process technology, most of works recently focus on integrating information retrieval and speech process, called SDR (Spoken Document Retrieval)[1][2][3]. In general, the methodology of SDR comprises mainly two levels, speech recognition and information retrieval. In the first level, queries and spoken documents are translated into a series of symbols, such as words, syllables, or phonemes. In the second level, the query transcripts are retrieved from those of spoken documents, where a lot of related information retrieval techniques can be utilized such as confusion matrix, document expansion and prosodic information.

However, there are still some researches about speech information retrieval without large-vocabulary recognizer [4] [5]. In [4], they implemented a system for user to search keywords in the speech database. In [5], a dynamic programming method was proposed to find the most frequent speech segments in the speech data and these segments were taken as indices to summarize the speech data.

As to MPEG-7 [6], officially called "Multimedia Content Description Interface", it defined lots of LLDs for multimedia. Audio LLDs are descriptors for audio data. There are two parts of MPEG-7 audio LLDs, audio spectrum descriptors and timbre descriptors. Audio spectrum descriptors describe several kinds of spectrum features of each audio frame, such as Spectrum Envelope, Centroid, Spread, and Flatness, etc. Timbre descriptors describe features of an entire audio segment, such as Harmonic spectral descriptors, SpectralCentriod, LogAttackTime, TempralCentroid, etc.

Most of the works about SDR [1] [2] [3] are engaged in retrieving data from a large amount of spoken data, such as broadcast news, TV programs, or speech recordings, etc. In [1], a database of 757 news recordings was considered. In [2] and [3], their databases take about 10 hours of speech signal. Although these systems have been proven effective, they need to construct lots of acoustic models and the language model. There is no denying that their computational complexity and memory requirement are very high. These stern conditions make them are not suitable for some embedded systems, such as handheld devices, which are limited on low power consumption and small memory size. In order to overcome these problems, we propose a multi-level matching method without the speech recognizer that may contain recognition errors and symbol boundary matching problems. Besides, we adopt MPEG-7 LLDs as speech features to study their retrieval performance.

This paper is organized as follows. In Section 2, MPEG-7 audio LLDs are introduced. In Section 3, our proposed method is described in details. The experiments about the discriminating capabilities of MPEG-7 audio LLDs and retrieval result of our retrieval system are discussed in

Section 4. Finally, we give the conclusions and future works in Section 5.

2. MPEG-7 AUDIO LOW-LEVEL DESCRIPTORS

2.1 Feature Descriptions

We adopt some of the audio low-level descriptors in MPEG-7 as our speech features. MPEG-7 audio low-level descriptors consist of a collection of simple, low complexity descriptors that are categorized into two parts, audio spectrum descriptors and timbre descriptors. We introduce the descriptors adopted in the paper as follows.

- **Audio spectrum centroid (ASC)** describes the center of gravity of the log-frequency power spectrum.
- **Audio spectrum spread (ASS)** describes the second moment of the log-frequency power spectrum.
- **Audio spectrum flatness (ASF)** describes the flatness properties of the spectrum of an audio signal within a given number of frequency bands.
- **Instantaneous harmonic spectral centroid (IHSC)** describes the amplitude weighted mean of the harmonic peaks of the spectrum.
- **Instantaneous harmonic spectral spread (IHSS)** describes the amplitude weighted standard deviation of the harmonic peaks of the spectrum, normalized by the harmonic spectral.

2.2 Computation Comparison between MPEG-7 LLDs and MFCCs

In this section, we analyze the computation complexity of spectrum and instantaneous harmonic descriptors (ASC+ASS+ASF+IHSC+IHSS) on the numbers of addition /subtraction, multiplication, division per frame. Furthermore, we compare these computational overloads with MFCCs [8] for one frame (240 samples, 8k sampling rate) From Table 2.1, we can find the low-complexity advantage of MPEG-7 LLDs.

Table 2.1 The Computation Loads of Extracting MPEG-7 LLDs and MFCCs for one frame

(240 samples, 8k sampling rate)

	Add./Sub.	Mult.	Div.
Total computations of the MPEG-7 LLDs	909	354	41
MFCCs	2768	2800	0

3. SYSTEM ARCHITECTURE OVERVIEW

An overview of our proposed system is depicted in Fig. 3.1, where the rectangular blocks stand for operation procedures.

We divide the retrieval processes into four procedures as follows. :

1) Audio/speech features extraction, in which MFCCs and MPEG-7 low-level descriptors are extracted.

2) **Similar frames tagging** for marking similar frames in database with query.

3) Possible segments extraction for verifying candidates corresponding to query.

4) Ranking these candidates and retrieve top N ones.

For all procedures above, we will describe them in the followed sections.

3.1 Speech feature extraction

For each frame of the speech data, we extract features including 13 MFCCs, 1 ASC, 1 ASS, 15 ASF, 1 IHSC, and 1HSS in the case of 8K sampling rate. The size of a frame is 30 ms (240 samples), while a frame is extracted with Hamming window-weighting every 10 ms (80 samples). These basic specifications are list in.

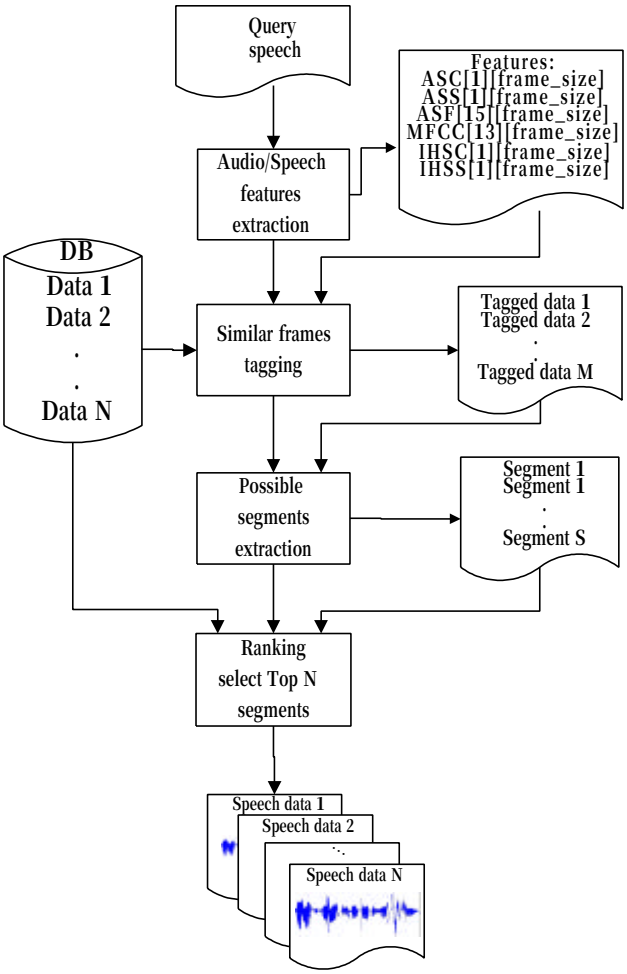


Figure 3.1 Overview of our retrieval system.

Table 3.1 Specifications of Feature Extraction

Sampling Rate	8 K/sec					
Frame Size	240 samples (30 ms)					
Frame Overlap	160 samples (20 ms)					
Features	ASC	ASS	ASF	IHSC	IHSS	MFCC
Feature Dimension	1	1	15	1	1	13

3.2 Similar frames tagging

It is a big overload for direct matching every frame interval. Therefore, we roughly tag the similar frames in speech data for help to allocate the possible segments. At first, we divide the query into several blocks with the length of 3 frames. The reasons we take a block as a processing unit are for reducing computations and no losing perception. Then, we calculate, block by block, the feature distance between the query and the sentences in database. If the distance is below our estimated threshold, $\boldsymbol{th_1}$, the block will be tagged with 1, otherwise 0. After calculating all block distances between query and each sentence, there will be an output data, ***tagged data*** for each sentence. In the same way, we get another ***tagged data*** by using the other features. Finally, we add those ***tagged data*** for different features as the ***final tagged data***.

3.3 Possible segment extraction

Here, we will extract several possible segments of speech data in the database based on the ***tagged data*** got in section 3.2. At first, we check the segments in each data if they are ***high-tagged***. Then, these ***high-tagged*** ones are extracted as candidates corresponding to the query.

Our implementation idea to check ***high-tagged*** segments is to utilize hamming window scanning method. First, we do convolution of ***tagged data*** with a Hamming window of query length to get a new data, $array_m$. And then, each local maximal value in $array_m$ is checked if it is greater than the threshold, th_2 for ***high-tagged***. These ***high-tagged*** segments, whose centers are the local maxima, are extracted. Finally, we check the next ***tagged data*** till check all of them. Fig. 3.2 shows the process of section 3.2 and section 3.3.

3.4 Ranking sentences output

After extracting these possible segments, we rank them by utilizing the DTW (Dynamic Temporal Warping) algorithm. Based on each feature, we can get the matching results and then we adopt the rank-based method, as described in (3.1), to integrate them. In (3.1), Rank is final rank output, while *i*, *j*, *f* are the indices for queries, data, descriptors respectively, and w_f is the weight for different features.

$$Rank(q_i, d_j) = \sum_{f=1,2,3,....} w_f \, rank_f(q_i, d_j) \qquad (3.1)$$

For example, we consider a segment whose rank for ASC is 3 while 5, 2, and 1 for ASS, ASF, and MFCC respectively. We will get a final rank, 1.68 by (3.1), where weight is 0.05, 0.05, 0.29, and 0.7.

Finally, according to the ranks of possible segments, the system outputs the sentences corresponding to the top N segments to the user. Fig. 3.3 displays the process of ranking possible segments and outputting corresponding sentences.

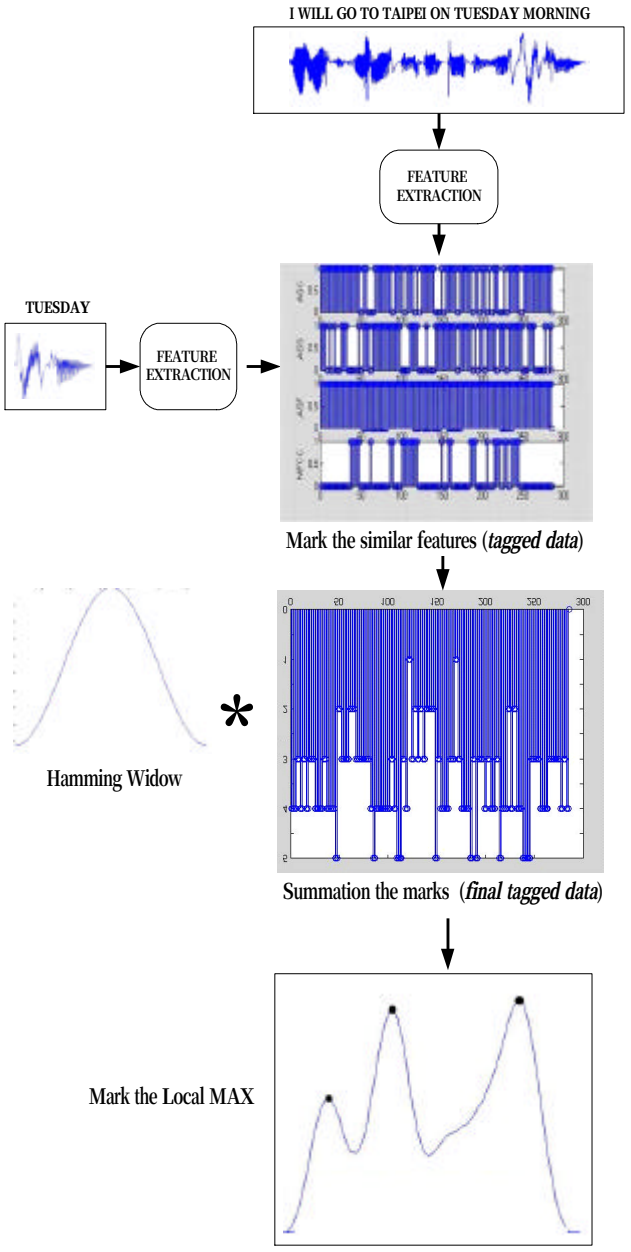


Figure 3.2 Process of similar frame tagging and possible segment extraction

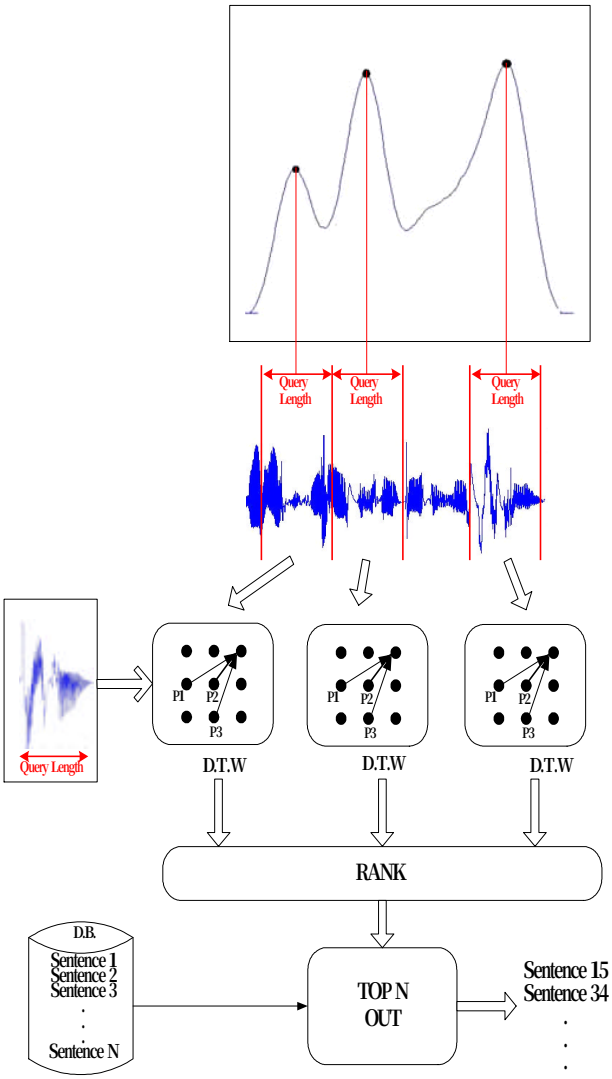


Figure 3.3 Process of ranking and output

3.5 Computational analysis of the direct matching method and our two level matching method.

In this section, we compare the computational load of our proposed method with direct matching method, which applies the DTW matching every frame interval [4] [5].

3.5.1 The direct matching method

We analyze the computation complexity from two main processes of DTW. First, we calculate the distance between frames of the query and the sentence. Secondly, we select the path whose cumulative distance is the shortest.

3.5.1.1 Local distance

Parts of total computation about local frame distance depends on the feature dimension, so we use ***O(local_dist_add)*** and ***O(local_dist_mul)*** to present the computation of additions and multiplications.

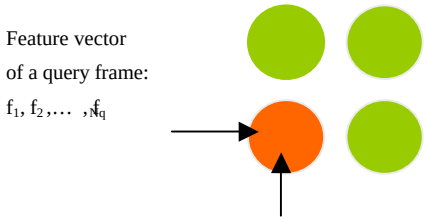
As shown in Fig. 3.4, N_q and N_d are frame numbers of query and sentence respectively. We need to compute the local distance N_q^2 times in one interval and the total shift number is about N_d-N_q. Therefore, the total computation loads are

Additions:

$N_q^2*(N_d-N_q)*O(local\_dist\_add)$, and

Multiplications:

$N_q^2*(N_d-N_q)*O(local\_dist\_mul)$



$$D = \sum_{i=1}^{N} (f_i - f_i')^2$$

N is the dimension of feature vectors

Figure 3.4 The Computation of Local Frame Distance

3.5.1.2 Path selections

As to the path selection, the type of DTW is shown in Fig. 3.5. It takes 4 additions,

3 addition operations to decide which is the last node and

1 addition to accumulate the path distance.

The same as above; the total counts for path selection are $N_q^2*(N_d-N_q)$. Since, the DTW path constraint factor is about **0.3**. Therefore, this part takes additions: $1.2*N_q^2*(N_d-N_q)$

$$m(k) = \min\left[f_x(k) - f_x(k-1), f_y(k) - f_y(k-1)\right] \qquad (3.2)$$

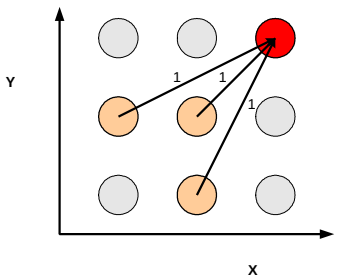


Figure 3.5 Path Selections

3.5.2 Our method

3.5.2.1 Similar frame tagging

We divide the query into several blocks with length of ***blocksize*** (3 frames) and compute local distance, block by block, and the shift interval is 2 frames, (N_{shift}).

The numbers of blocks are about $N_q/blocksize = N_q/3$.

Besides, each block needs S_N times for local distance computations,

$S_N = blocksize*(N_d - blocksize)/N_{shift}$

$= 3*(N_d-3)/2 \sim= 1.5*N_d$.

Therefore, we needs

Additions: $0.5*Nd*Nq*O(local\_dist\_add)$.

Multiplications $0.5*Nd*Nq*O(local\_dist\_mul)$.

3.5.2.2 Possible segment extraction

The main computation loads of this step are convolution between tagged data and a hamming window ($l_w = N_q$) .It takes about

Additions: N_q*N_d, and

Multiplications: N_q*N_d

3.5.2.3 Possible segment ranking

This part depends on the number of possible segments, $N_{possibleseg}$ that we extract for DTW. So it takes about

Additions:

local distance: $N_q^2*N_{possibleseg}*O(local\_dist\_add)$*+ path selections :* $1.2*N_q^{2*}N_{possibleseg}$

Multiplications:

local distance: $N_q^2*N_{possibleseg}*O(local\_dist\_mul)$.

3.5.3 Computational comparisons

For section 3.5.1 to 3.5.2, the totally analytic results are concluded in Table 3.2. As shown in Table 3.3, we see that the average numbers of possible segments, $N_{possibleseg}$, are usually far less than the average frames of sentences, N_d. Therefore, from Table3.2 and Table 3.3, since $N_q << N_d$, we find that the additions and multiplications are dominated by $N_q*N_d*O(local\_dist)$, while the parts of the direct matching (D.M.) method are dominated by $N_q^2*N_d*O(local\_dist)$. Comparing with the direct matching method, our method is about $1/N_q$ times of the computational load of them.

4 EXPERIMENT RESULTS

We perform the experiments to evaluate the retrieval performance of our proposed system. The object speech database in these experiments consists of 100 sentences (50 oral sentences and 50 news titles). Each data is uttered by a single person and is recorded twice. The sampling frequency is 8 KHz and the frame interval is 10 ms. MFCCs, ASC, ASS, ASF, IHSC, and IHSS are the features extracted from each frame. We use 50 spoken keywords as inputs totally and check if the retrieval spoken documents include them. Besides, we show the retrieval performance of the direct matching method as the baseline in the followed experiments.

Table 3.2 The Computation Loads of Our Method and the Direct Matching Method.

(The dominated terms are shown in boldface.)

	Add	Mul
D.M. [4],[5]	$N_q^2*(N_d-N_q)*$ $(1.2+O(local\_dist\_add))$	$N_q^2*(N_d-N_q)*O(local\_dist\_mul)$
Ours	$0.5*N_q*N_d*O(local\_dist\_add)$ $+N_q*N_d$ $+(1.2+0.3*N_q^2*N_{possibleseg})*$ $O(local\_dist\_add)$	$0.5*N_q*N_d*O(local\_dist\_mul)$ $+N_q*N_d$ $+0.3*N_q^2*N_{possibleseg}$ $*O(local\_dist\_mul)$

Table 3.3 The Average Numbers of Query Frames, Sentence Frames and Possible Segments

Avg. Nq	70.64
Avg. Nd	345.23
Avg. Npossibleseg	25.43

Table 4.1 shows the average retrieval accuracy and operational analysis by using a single feature. ASF has the best retrieval effects of the adopted MPEG-7 LLDs, while retrieval accuracy of the others are under 0.4. Additionally, the gaps between ASF and MFCC are about 0.03.

Table 4.2 shows the average retrieval accuracy and operational analysis by using the combination of features. We consider the combination of ASx (ASC+ASS+ASF), IHSx (IHSC+IHSS), LLDs (ASx+IHSx), and ALL (LLDs and MFCC). Besides, the performance of the combination of MPEG-7 LLDs meets 77.51%.

Comparing to the direct matching method, our proposed method degrades about 3% except instantaneous harmonic features. The decline is mainly from the extraction of possible segments. It may lose some possible segments else. However, as mention in 3.5, our matching method can reduce the computation overload greatly. Therefore, our method can provide a faster and little-degrading method.

5 CONCLUSIONS

This paper proposes a spoken sentence retrieval system. Instead of using traditional large-vocabulary recognizer, we adopt a two-level retrieval method to reduce the computational complexity. Besides, we choose the MPEG-7 audio LLDs as our features. They were proven comparable to the MFCC in our experiment result. Furthermore, the combination of MPEG-7 LLDs and MFCC has better performance than using them alone.

Table 4.1 The average retrieval accuracy and operational analysis by using a single feature between our method and direct matching method.

	ASC	ASS	ASF	IHSC	IHSS	MFCC
Ours.	0.413 /19 /76	0.391 /24 /83	0.760 /1.7 /74	0.220 /0.13 /0.3	0.301 /0.2 /1	0.7943 /4.7 /12
D.M.	0.420 /1923 /2081	0.427 /2438 /2254	0.809 /172 / 2004	0.441 /13 /7	0.577 /19 /27	0.821 /468 /340

Accuracy/Million Additions/Million Multiplications

Table 4.2 The average retrieval accuracy and operational analysis by using combination of features between our method and direct matching method.

	ASx	IHSx	LLDs	MFCC	ALL
Ours.	0.7599 /45.3 /233	0.3067 /0.336 /1.3	0.7751 /45.636 /234	0.7943 /4.7 /12	0.8076 /50 /246
D.M.	0.7687 /4532 /6340	0.5233 /32 /34	0.7979 /4564 /6374	0.821 /468 /340	0.8339 /4632 /6714

Accuracy/Million Additions/Million Multiplications

REFERENCES

[1] Berlin Chen; Hsin-min Wang; Lin-shan Lee; "Discriminating capabilities of syllable-based features and approaches of utilizing them for voice retrieval of speech information in Mandarin Chinese", Speech and audio Processing, IEEE Transactions on , Volume: 10 Issue: 5 , Jul 2002, Page(s): 303 -314

[2] Matthew A. Siegler, "Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance", Electrical and Computer

Engineering Carnegie Mellon University Pittsburgh, Pennsylvania 15213, 1999 December 15

[3] Ng, K, Zue, VW, "Phonetic recognition for spoken document retrieval", Acoustics, Speech, and Signal Processing, 1998. ICASSP'98. Proceedings of the 1998 IEEE International Conference on, Volume: 1 , 12-15 May 1998, Page(s): 325 -328 vol.1

[4] H.K Xie, "A Study on Voice Caption Search for Arbitrarily Defined Keywords." Master Thesis, National Taiwan University of Science and Technology, Taiwan, R.O.C., July 2000.

[5] Itoh, Y, "A matching algorithm between arbitrary sections of two speech data sets for speech retrieval"; Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on, Volume: 1 , 2001 , Page(s): 593 -596 vol.1

[6] "ISO/IEC FDIS 15938-4 Multimedia Interface Description Interface Part 4 Audio"

[7] Richard.B, Berthier.R"Modern Information Retrieval", New York: ACM Press, 1999

[8] Y. S. Weng, "The chip design of Mel frequency cepstrum coefficient for HMM Speech Reconition," Master Thesis, National Cheng Kung University, Taiwan, R.O.C., June 1998.