

Personal Spoken Sentence Retrieval Using Two-Level Feature Matching and MPEG-7 Audio LLDs*

PO-CHUAN LIN[†], JHING-FA WANG, JIA-CHING WANG AND JUN-JIN HUANG

[†]*Department of Electronics Engineering and Computer Science*

Tung Fang Institute of Technology

Kaohsiung, 829 Taiwan

Department of Electrical Engineering

National Cheng Kung University

Tainan, 701 Taiwan

Conventional spoken sentence retrieval (SSR) relies on a large-vocabulary continuous-speech recognition (LVCSR) system. This investigation proposes a feature-based speaker-dependent SSR algorithm using two-level matching. Users can speak keywords as the query inputs to get the similarity ranks from a spoken sentence database. For instance, if a user is looking for a relevant personal spoken sentence, "October 12, I have a meeting in New York" in the database, then the appropriate query input could be "meeting", "New York" or "October". In the first level, a Similar Frame Tagging scheme is proposed to locate possible segments of the database sentences that are similar to the user's query utterance. In the second level, a Fine Similarity Evaluation between the query and each possible segment is performed. Based on the feature-based comparison, the proposed algorithm does not require acoustic and language models, thus our SSR algorithm is language independent. Effective feature selection is the next issue in this paper. In addition to the conventional mel frequency cepstrum coefficients (MFCCs), several MPEG-7 audio low-level descriptors (LLDs) are also used as the features to exploit their ability for SSR. Experimental results revealed that the retrieval performance using MPEG-7 audio LLDs was close to that of the MFCCs. The combination of MPEG-7 audio LLDs and the MFCCs could further improve the retrieval precision. Based on the feature-based matching, the proposed algorithm has the advantages of language independent and speaker dependent training free. Comparing to the original methods [10, 11], with only 0.026 ~ 0.05 precision decrease, the addition and multiplication numbers are reduced by around a factor of l_q (frame number of query). It is particularly suitable for the use in resource-limited devices.

Keywords: audio low level descriptors, matching algorithm, MPEG-7, spoken sentence retrieval, feature-based comparison

1. INTRODUCTION

Spoken language is undoubtedly the most natural and convenient way for people to express and transmit their thoughts. With more requirements to access spoken data, an efficient retrieval method is essential. Most research on spoken data retrieval has focused on PC-based platforms. Methodologies for PC-based spoken sentence retrieval (SSR) [1-6] generally include two steps: speech recognition and information retrieval. During the first step, both spoken queries and spoken documents in the database are transcribed into

Received June 27, 2007; revised November 6, 2007; accepted January 31, 2008.

Communicated by Liang-Gee Chen.

* The paper has been presented in the 8th Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS 2003), Sydney, Vol. 12, 2003, pp. 9-14, sponsored by the Ministry of Economic Affairs, Department of Industrial Technology of the Taiwan, R.O.C.

a series of semantic units, such as phrases or syllables. In the second step, the query transcripts are used to retrieve the relevant spoken document transcripts using information retrieval techniques [7-9]. Although these retrieval systems have achieved a certain degree of success, these systems are constructed based on an acoustic model and a language model. The memory requirement is quite large for the resource-limited devices.

With the increasingly widespread use of personal portable device, it is useful to devise an efficient method for SSR in resource-limited devices without using an acoustic model or a language model. However, the speaker-dependent property is the limitation of the assumption at the basis of the feature level SSR. An SSR in personal devices commonly involve only personal spoken database for applications focusing on retrieving previous recorded sentences such as a voice memorandum and a voice phonebook. Therefore, it is suitable for this article to develop a speaker-dependent SSR system based on matching from speech feature. In general, people are often concerned about the not-so-great performance of a feature-based SSR as compared to an SSR that works at model-based approach. For an SSR used in a personal mobile information access environment and for achieving an acceptable retrieval performance for a feature-based SSR, we focus our work on a medium-sized (approximately 100 database sentences) database.

Effective representation of speech waveform is another crucial issue in an SSR system. This process converts the speech signal into parameters while virtually preserving the speech signal information. In the last decade, mel frequency cepstral coefficients (MFCCs) have drawn most attentions and been applied in many speech recognition systems. Nevertheless, the investigations seeking for other possible alternatives still continue. Moving picture experts group standard 7 (MPEG-7), formally named "multimedia content description interface (MSDI)", is established for describing the multimedia content data [12]. MPEG-7 aims to make multimedia data more searchable. Some example application areas for MPEG-7 audio are setting-up audio archives (radio, movies, TV), retrieving audio files from the Internet, filtering audio broadcasts, music education, and surveillance [21]. This paper evaluates the performance of several MPEG-7 audio low-level descriptors (LLDs) in our algorithm. Experimental results revealed that the retrieval performances of the adopted MPEG-7 audio LLDs and the MFCCs are similar. Furthermore, the combination of MPEG-7 audio LLDs and the MFCCs provides better retrieval performance than the individual MFCCs.

This paper is organized as follows. Section 2 describes the MPEG-7 audio LLDs adopted in this paper. The proposed two-level feature algorithm is detailed in section 3. The computational loads of the direct matching [10, 11] and the proposed methods are also analyzed in this section. In section 4, the evaluation to choose effective MPEG-7 audio LLDs as speech features is given first. The retrieval experimental results using single and multi-feature are then presented. Finally, section 5 draws the concluding remarks.

2. FEATURE EXTRACTION

2.1 MPEG-7 Audio Low-Level Descriptors

The audio descriptors are the basic components constituting the MPEG-7 audio standard. MPEG-7 audio LLDs consist of a collection of simple, low-complexity audio spectrum descriptors and timbre descriptors [13]. Audio descriptors are instantiations of

meta-data that may be associated with a single temporal interval or with a set of intervals in an acoustic waveform [13]. Although the MPEG-7 audio LLDs are mainly used to describe audio signals, they provide abundant information to portray speech signals. As no works have ever tried to adopt MPEG-7 audio descriptors in SSR, this motivates the authors to seek the possibility of using the MPEG-7 audio LLDs as the speech features for the proposed SSR algorithm. For SSR, this paper evaluates some audio spectrum descriptors including the audio spectrum centroid (ASC), the spread (ASS), the flatness (ASF), and the envelope (ASE) for describing many types of spectral features of a speech frame. Besides, some timbre descriptors including the harmonic spectral centroid (IHSC), the spread (IHSS), the deviation (IHSD) and the variation (IHsv) are also applied to describe an entire speech segment. Some evaluations (discussed in section 4) were made to select suitable descriptors for the proposed retrieval system. Moreover, a comparative performance evaluation between MFCCs and MPEG-7 audio LLDs was conducted in this paper. Finally, the retrieval performance of hybrid MFCCs and MPEG-7 audio LLDs was also evaluated in our experiments.

3. SPOKEN SENTENCE RETRIEVAL

3.1 Proposed Two-Level Matching Algorithm

An overview of our proposed system is depicted in Fig. 1, where the rectangular blocks stand for operation procedures. We divide the retrieval processes into three main steps as follows.

Step 1: MPEG-7/MFCC feature extraction

For each speech frame, we extract 33-dimension feature including: 13 MFCCs + 1 ASE + 1 ASC + 1 ASS + 15 ASF + 1 IHSC + 1 HSS in the case of 8 kHz sampling rate. The frame size is 30 ms (240 samples) with 20ms (160 samples) overlap, while each frame is extracted with the Hamming window-weighting.

Step 2: Possible segment extraction level

First, the *Similar Frame Tagging* is performed. Second, a *Highly Possible Segment Extraction* is designed with a rectangular scanning or a hamming window convolution to identify the possible segments that are similar to user's query.

Step 3: Fine similarity evaluation level

The *DP* distance between query and every possible segment extracted from **step 2** is calculated. The reciprocal of minimum *DP* distance is chosen as the similarity between query and a database sentence. Finally, we rank the possible segments based on the similarity and retrieve the top *M* ones. All of the procedures above will be described in the following sections.

3.1.1 Possible segment extraction level

The possible segment extraction level contains two sub processes, the *Similar Frame Tagging* and the *Highly Possible Segment Extraction*.

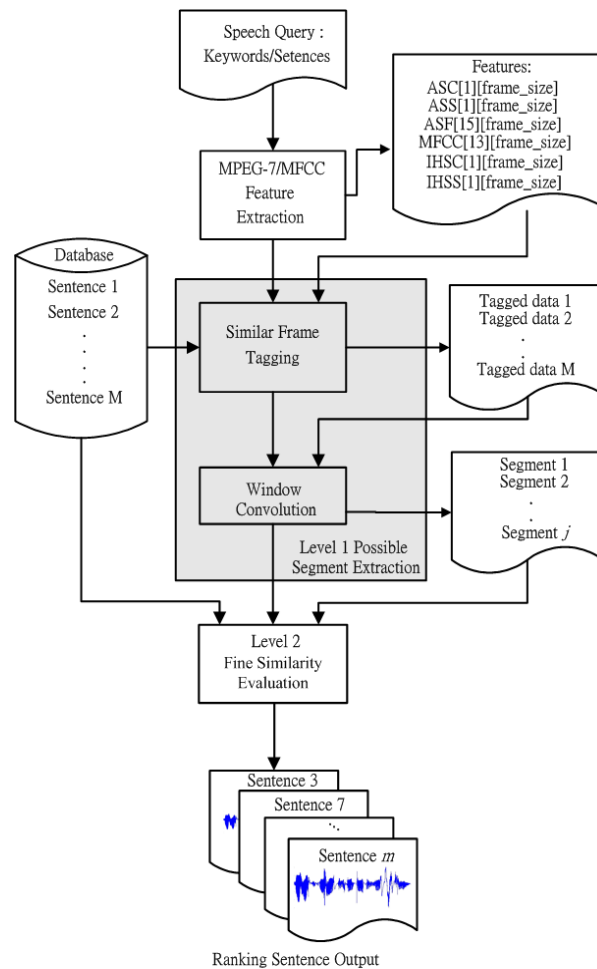


Fig. 1. Retrieval process of the proposed system.

(a) Similar frame tagging

The *Similar Frame Tagging* is performed to locate possible database segments which are similar to user's query. In Fig. 2, user's query and a database sentence are divided into speech blocks; each block contains three frames with one frame overlapping. For each query block, the *block distance* (*total distance of three consecutive speech frames*) between query and database sentences are calculated, if the *block distance* is smaller than an empirical threshold, th_1 , then the three frame-indexes of this block are tagged as 1; otherwise they are tagged as 0. Then, the query block shifts right two frames to the next database block. For every query block, this fashion is repeated until one query block reaches the end of a database sentence. The *Similar Frame Tagging* process is applied for each speech feature, and we call each binary stream as the *Block Tagged Data*. Finally, all the *Block Tagged Datas* are summed up to yield the *Tagged Data*. Fig. 3 gives an example to illustrate the *Similar Frame Tagging* with using four features: ASC, ASS, ASF and MFCCs.

with the size of query length l_q . The i and j represent the frame index and possible segment index of a database sentence. If the summation of values from i to $I + l_q - 1$ of *Tagged_Data* is greater than a threshold th_2 , we consider this highly tagged segment as a possible segment. Normally, the *hop_size* for window shifting is set as one. To avoid over-extracting neighbor segments, the *hop_size* is changed as $0.5 l_q$ right after a possible segment is detected. The pseudo code for extracting possible segment using rectangular window scanning is described as follows.

```

i = 0; j = 0; /* Initialization */
while i <  $l_d$ 
   $array[i] = \sum_{x=i}^{i+l_q-1} Tagged\_Data[x];$ 
  if  $array_m[i] > th_2$ 
     $S_j = data_m[i : i + l_q - 1];$ 
    j = j + 1;
    hop_size =  $0.5 * l_q$ ;
  else
    hop_size = 1;
  end if;
  i = i + hop_size;
end while;
Finish: Output all possible segments  $S_j$ ;

```

(b.2) Highly possible segment extraction-using hamming window convolution

In this part, the window scanning is accomplished by signal convolution. We adopt a Hamming window with the size of query length l_q , and convolute this window with the *Tagged_Data*. The pseudo code for extracting possible segment using the Hamming window convolution is given below:

```

i = 0; j = 0; /* Initialization */
Step 1: /* Convoluting Tagged_Data and Hamming window function  $f(x)$  */
for each frame i in the Tagged_Data
   $array[i] = \sum_{x=-l_q/2}^{l_q/2} f(x) \cdot Tagged\_Data[i - x];$ 
end for;
go to step 2;

Step 2: /* Possible segment identification */
for each local maximal of  $array[i_{local\_max}]$ 
  if  $array[i_{local\_max}] > th_3$ 
     $S_j = data[i_{local\_max} - l_q/2 : i_{local\_max} + l_q/2 - 1];$ 
    j = j + 1;
  end if;
end for;
Finish: Output all possible segments  $S_j$ ;

```

In step 1, *Tagged_Data* is convoluted with a Hamming window. Step 2 is used to identify the highly tagged segment as a possible segment or not. In step 2, each local maximum value $array[i_{local_max}]$ is considered if it is greater than a threshold th_3 . If it is, a segment S_j (length = l_q and center frame index = i_{local_max}) is labeled as a possible segment. The overall concept of possible segment extraction level is illustrated in Fig. 4.

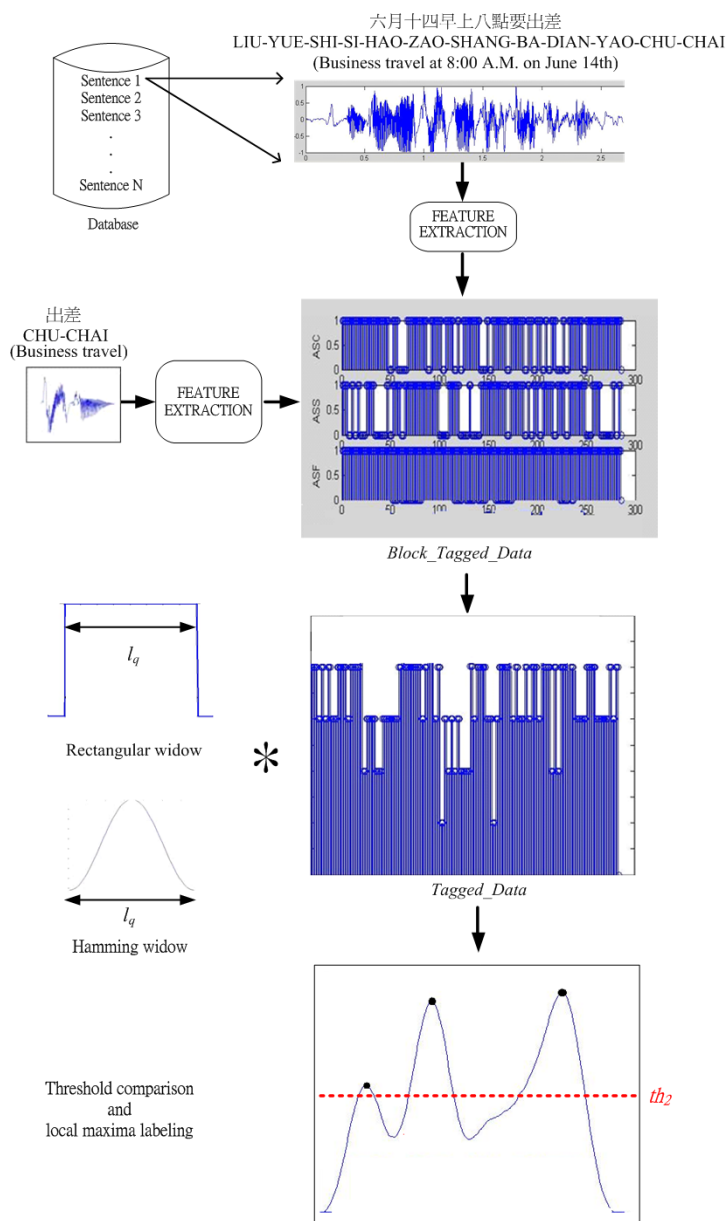


Fig. 4. The overall concept of possible segment extraction level.

3.1.2 Fine similarity evaluation level

After the possible segments have been extracted, the fine similarities between query and these possible segments are calculated by the dynamic programming (*DP*) algorithm. Assume there are M spoken sentences in the database. Let $score(query, possible_segment_k^m)$, $m = 1 \sim M$ denotes the matching score of query and k th possible segment in m th database sentence. For multiple feature set $f = 1 \sim F$, this matching score is calculated by:

$$score(query, possible_segment_k^m) = \sum_{f=1}^F w_f \cdot DP(query(f), possible_segment_k^m(f)), \quad (1)$$

where $DP(query(f), possible_segment_k^m(f))$ is the accumulated distance between query and k th possible segment in m th database sentence using f th feature, and w_f is weighting factor for different speech features. The matching score between query and m th database sentence is determined by

$$score(query, sentence^m) = \min_k(possible_segment_k^m). \quad (2)$$

As the matching score is obtained based on the *DP* distance, the smaller matching score indicates the higher similarity. Finally, the system ranks all the database sentences in accordance with these similarity matching scores. The overall concept of the fine similarity evaluation level is illustrated in Fig. 5.

3.2 Computational Analysis

3.2.1 Direct matching method

This section compares the computational load of the proposed algorithm with that of the direct matching method [10, 11], which applies *DP* algorithm to every frame interval. For direct matching, the computational complexity analysis includes: (a) the local distance calculation between query frame and the database sentence frame; (b) the shortest path selection of *DP* algorithm.

(a) Local distance

The local distance indicates the frame-difference between r th frame of user's query (reference pattern) and s th frame of a database sentence (unknown utterance) on the *DP* plane. The computation of a local frame distance depends on speech feature selection and its dimension; $\Phi(local_add)$ and $\Psi(local_mul)$ represent the addition and multiplication times of different features. As indicated in Fig. 6, l_q and l_d denote the frame numbers of query and database sentence, respectively. The local distance must be computed l_q^2 times per frame interval; the total number of shifts is approximately $(l_d - l_q)$. Therefore, the total computational loads are:

$$\#Additions: l_q^2 \cdot (l_d - l_q) \cdot \Phi(local_add) \cong l_q^2 \cdot l_d \cdot \Phi(local_add). \quad (3)$$

$$\#Multiplications: l_q^2 \cdot (l_d - l_q) \cdot \Psi(local_mul) \cong l_q^2 \cdot l_d \cdot \Psi(local_mul). \quad (4)$$

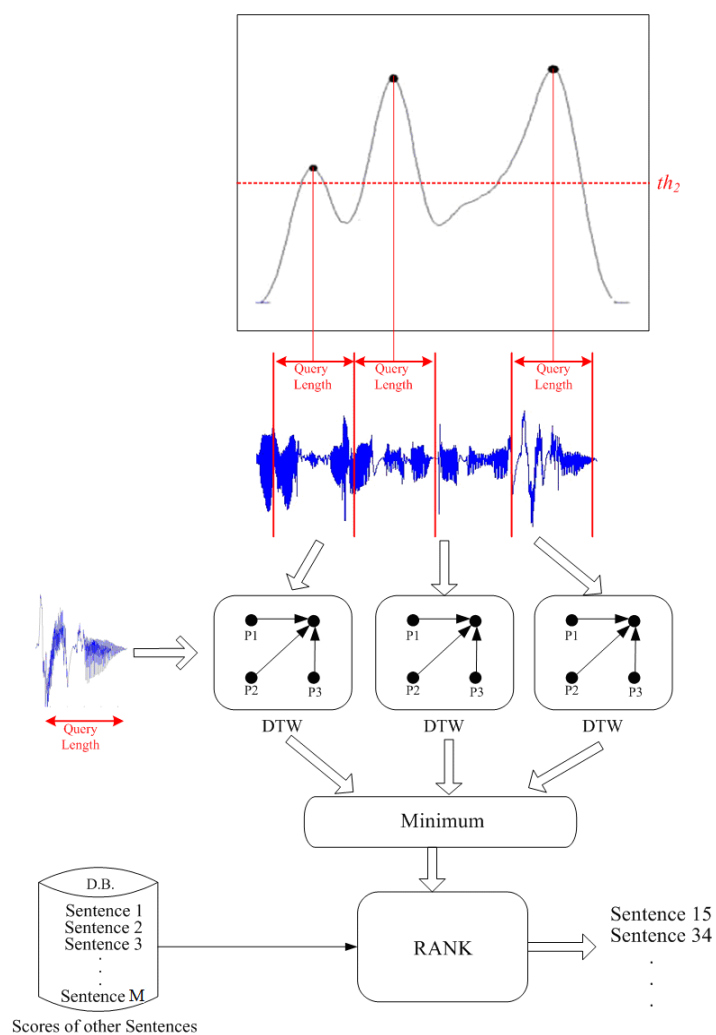


Fig. 5. The overall concept of fine similarity evaluation level.

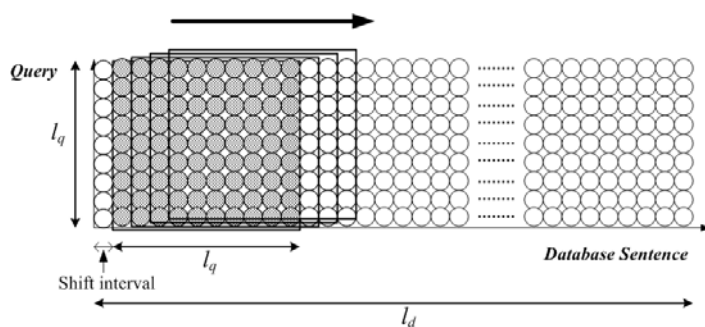


Fig. 6. Illustration of the direct matching method.

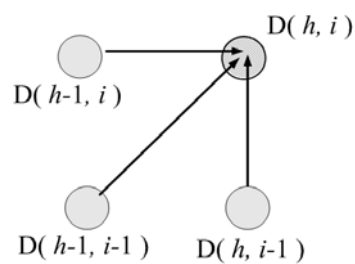
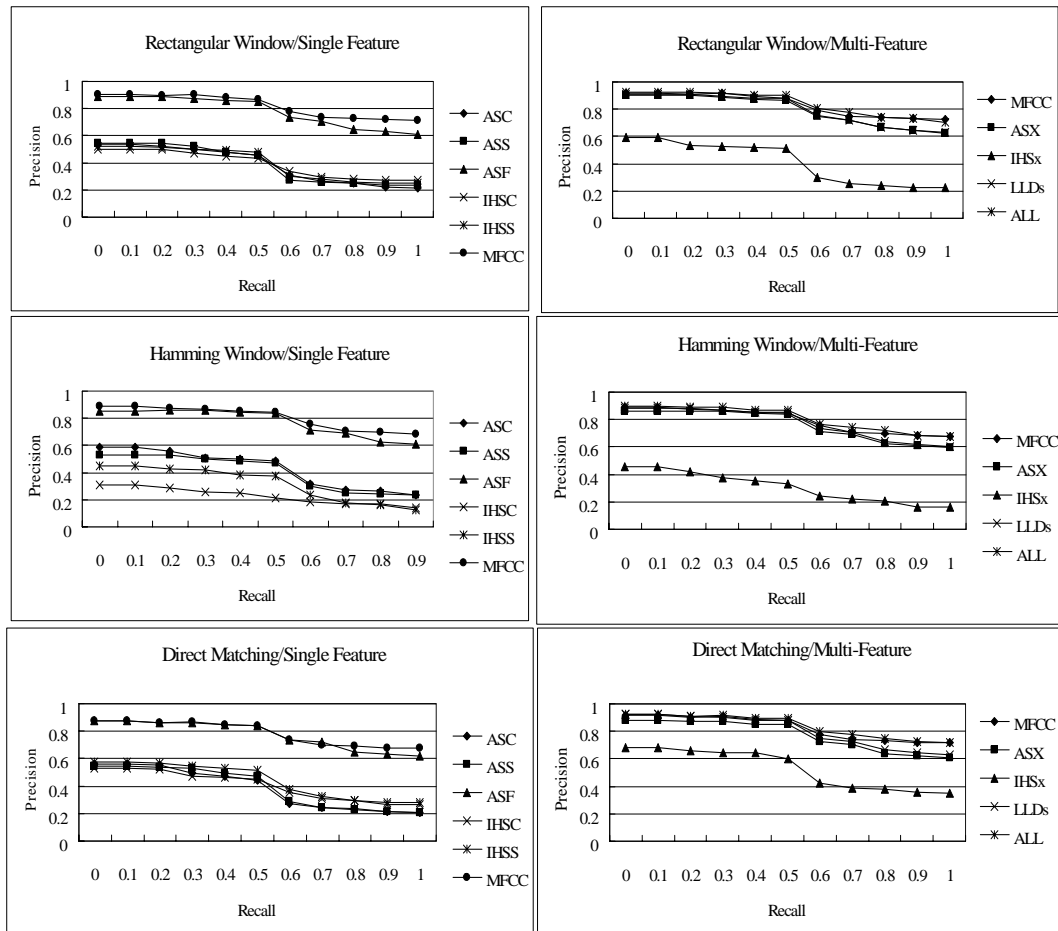


Fig. 7. Three directional path selections.

(b) Path selection

As indicated in Fig. 7, the horizontal axis represents the query frame and the vertical axis denotes the database sentence frame. For each node, the path decision needs four additions and three additions for last node determination; one addition for accumulating the path distance. As above, the total number of additions required for selecting a path is $4 \cdot l_q^2 \cdot (l_d - l_q)$. As shown in Fig. 8, the type one $0^\circ - 45^\circ - 90^\circ$ dynamic time warping (DTW) local path constraint is adopted [18]. For reducing the computational load, we utilize a global path constraint, which excludes certain part of the DTW nodes from the region the optimal path can traverse [16], *i.e.*, only the nodes (r, s) that satisfy the rule described in Eq. (5) will be considered in the search procedure.



(a) Single feature.

(b) Multi feature.

MPEG-7 ASx (spectrum descriptors): ASC, ASS, ASF; MPEG-7 IHSx (instantaneous harmonic descriptors): IHSC, IHSS; MPEG-7Audio LLDs: ASC, ASS, ASF, IHSC, IHSS.

Fig. 8. Experimental results of precision-recall curves.

$$|r - s| \leq W, \quad (5)$$

where W is an appropriate positive integer called the warping factor. This design parameter corresponds to the fact that time-axis fluctuation in usual cases never causes excessive timing difference [17]. A larger W indicates a larger possible searching area; this makes a higher percentage of possible warps to be performed [19]. However, the computational load is heavier. Therefore, it is a trade off problem between computational load and system performance. As the proposed algorithm, we need to evaluate the matching similarity between a possible segment and query with equal length. Moreover, the mapping path belongs to the type of ‘‘anchored beginning, anchored end’’. To limit the mapping path to be a reasonable path and reduce computational load, the warping factor W was set equal to 3, which covered the utmost ± 90 ms timing difference. By this setting, the corresponding global path constraint factor (the ratio of possible searching area to overall area) is approximately 0.6. Therefore, the path selection requires the following number of additions.

$$\text{\#Additions: } 4 \cdot (0.6) \cdot l_q^2 \cdot (l_d - l_q) = 2.4 \cdot l_q^2 \cdot (l_d - l_q) \cong 2.4 \cdot l_q^2 \cdot l_d. \quad (6)$$

3.2.2 Two-Level feature matching method

(a) Similar frame tagging

First, a query is divided into numerous blocks ($block_size = 3$ frames). Second, the local distance is computed, and query block shift right two frames ($N_{shift} = 2$) to the next database block. The total blocks of query N_{block}^q is approximately equal to $l_q/N_{shift} = l_q/2$. Therefore, each block requires S_N (number of right shifting for each block) times for local distance computation.

$$S_N = (l_d - block_size)/N_{shift} = (l_d - 3)/2. \quad (7)$$

The following numbers of additions and multiplications are required.

#Additions:

$$\begin{aligned} N_{block}^q \cdot block_size \cdot S_N \cdot \Phi(local_add) &= (l_q/2) \cdot 3 \cdot ((l_d - 3)/2) \cdot \Phi(local_add) \quad (8) \\ &\cong 0.75 \cdot l_q \cdot l_d \cdot \Phi(local_add). \end{aligned}$$

#Multiplications:

$$\begin{aligned} N_{block}^q \cdot block_size \cdot S_N \cdot \Psi(local_mul) &= (l_q/2) \cdot 3 \cdot ((l_d - 3)/2) \cdot \Psi(local_mul) \quad (9) \\ &\cong 0.75 \cdot l_q \cdot l_d \cdot \Psi(local_mul). \end{aligned}$$

(b) Possible segment extraction using rectangular/Hamming window scanning/convolution

As shown in the Fig. 4, the rectangular window (size = l_q) is sliding with the *Tagged_Data* (length = l_d), the *Tagged_Data* are summarized within a rectangular window.

$$\text{\#Addition} = \text{length of } Tagged_Data \cdot l_q = l_d \cdot l_q. \quad (10)$$

In the Hamming window convolution method, most of the computational load is associated with the convolution between the *Tagged_Data* and a Hamming window (size = l_q), which requires the following numbers of operations.

$$\#Additions = \#Multiplications \cong l_q \cdot \text{length of tagged_data} = l_q \cdot l_d. \quad (11)$$

(c) Matching possible segments with queries for ranking sentences outputs

The computational load of sentence ranking depends on the number of possible segments, $N_{possible_seg}$, which extracted for *DP* distance computing. It requires the following numbers of operations.

$$\begin{aligned} \#Additions \text{ (local distance):} \\ l_q^2 \cdot N_{possible_seg} \cdot \Phi(\text{local_dist_add}) + \text{path selection: } 2.4 \cdot l_q^2 \cdot N_{possible_seg}. \end{aligned} \quad (12)$$

$$\#Multiplications \text{ (local distance): } l_q^2 \cdot N_{possible_seg} \cdot \Psi(\text{local_dist_mul}). \quad (13)$$

Summing up the above, Table 1 lists all of the analytic results.

Table 1. The computational complexity of proposed methods and direct matching method (the dominated terms are shown in boldface).

Method	Number of additions	Number of multiplications
Rectangular window	$l_q \cdot l_d \cdot (1 + 0.75 \cdot \Phi(\text{local_add})) + l_q^2 \cdot N_{possible_seg} \cdot (2.4 + \Phi(\text{local_add}))$	$0.75 \cdot l_q \cdot l_d \cdot \Psi(\text{local_mul}) + l_q^2 \cdot N_{possible_seg} \cdot \Psi(\text{local_mul})$
Hamming window	$l_q \cdot l_d \cdot (1 + 0.75 \cdot \Phi(\text{local_add})) + l_q^2 \cdot N_{possible_seg} \cdot 2.4 + \Phi(\text{local_add})$	$l_q \cdot l_d \cdot (1 + 0.75 \cdot \Psi(\text{local_mul})) + l_q^2 \cdot N_{possible_seg} \cdot \Psi(\text{local_mul})$
Direct matching	$l_q^2 \cdot l_d \cdot (2.4 + \Phi(\text{local_add}))$	$l_q^2 \cdot l_d \cdot (2.4 + \Psi(\text{local_mul}))$

The mean value of possible segments, $N_{possible_seg}$, are usually far smaller than the average number of frames per sentences, l_d , so the proposed algorithm substantially reduces the number of *DP* matches. The direct matching method are dominated by $l_q^2 \cdot l_d \cdot \Phi(\text{local_add})$ and $l_q^2 \cdot l_d \cdot \Psi(\text{local_mul})$. Since $l_q \ll l_d$, the addition and multiplication number of the proposed algorithm are dominated by $0.75 \cdot l_q \cdot l_d \cdot \Phi(\text{local_add})$ and $0.75 \cdot l_q \cdot l_d \cdot \Psi(\text{local_mul})$, respectively. Compared with the direct matching method, the proposed algorithm reduces around a factor of l_q .

4. EXPERIMENTAL RESULTS

4.1 Individual Feature Evaluations by Exact Matching

This section evaluated the performance of MPEG-7 audio LLDs and MFCCs. Features with better performance for spoken sentence recognition were chosen for our two-level SSR system. For each dataset, the content of query was exactly the same with one of the database sentence, and was spoken by the same speaker, but uttered at another time. This experiment directly evaluated the *DP* distance between query and each database sen-

tence without steps 2 which discussed in section 3.1. In other words, we replace Eq. (1) as

$$\text{score}(\text{query}, \text{database_sentence}^m) = \sum_{f=1}^F w_f \cdot DP(\text{query}(f), \text{database_sentence}^m(f)). \quad (14)$$

The overall retrieval performance was evaluated based on the non-interpolated mean average precision (mAP) [14, 15]. The mAP is defined as follows:

$$\text{mAP} = \frac{1}{L} \sum_{i=1}^L \left\{ \frac{1}{M} \sum_{j=1}^{M_i} \left\{ \frac{1}{N_j} \sum_{k=1}^{N_j} \text{prec}N_{Q_j}(k) \right\} \right\}, \quad (15)$$

where N_j denotes the total number of relevant sentences for query j , M_i represents the total number of queries in batch i , L is the total number of query batches, and $\text{prec}N_{Q_j}(k)$ is the precision for query Q_j when k sentences are retrieved. Three types of spoken data: (1) names, (2) conversation sentences and (3) news titles were employed. Their average time durations were 1.2, 3.3, and 5.7 seconds, respectively. The experimental results are presented in Table 2.

Table 2. mAP evaluations for individual MPEG-7 audio LLDS and MFCCs.

Dataset	ASE	ASC	ASS	ASF	MFCC	IHSC	IHSS	IHSV	IHSD
25 names	0.422	0.483	0.359	0.782	0.873	0.539	0.544	0.193	0.352
50 names	0.250	0.295	0.223	0.424	0.579	0.326	0.315	0.101	0.179
25 conversation sentences	0.417	0.797	0.756	0.818	0.868	0.779	0.793	0.471	0.558
50 conversation sentences	0.366	0.796	0.739	0.816	0.860	0.774	0.793	0.463	0.510
25 news titles	0.539	0.949	0.824	0.900	0.981	0.959	0.928	0.567	0.639
50 news titles	0.450	0.889	0.782	0.893	0.971	0.929	0.904	0.509	0.575

As shown in Table 2, the MFCCs gave the best matching performance, and ASF was better than the other LLDS, except news titles, in which case IHSC and IHSS outperform ASF. Moreover, for a larger database, the matching performance was degraded. The contents between query and database sentences became more diverse as the size of data set increased. Finally, the ASC, ASS, ASF, IHSC and IHSS were chosen to be combined as the feature set for the following experiments.

4.2 Spoken Sentence Retrieval

The spoken sentence database consisted of 100 Mandarin sentences (50 personal schedules and 50 news titles) in the following experiments. Each utterance was uttered by a single person. Besides, the retrieval performance of the direct matching method was taken as the baseline result. Table 3 specifies the statistics of the speech query and database used in our experiments.

Table 3. Statistics of the speech query and database.

Number of spoken sentences to be retrieved	100		
Number of queries	50		
	Min	Max	Mean
Database sentence length (sec)	3.3	7.6	5.4
Query length (sec)	0.4	1.3	0.8
Number of relevant sentence/query	1	10	2.7

The most commonly used measures-recall rate and precision rate-were used [20] to evaluate the performance of retrieval systems, and were defined by Eqs. (16) and (17), respectively.

$$\text{Recall Rate} = \frac{\text{number of relevant records retrieved}}{\text{total number of relevant records in collection}}, \quad (16)$$

$$\text{Precision Rate} = \frac{\text{number of relevant records retrieved}}{\text{total number of records retrieved}}. \quad (17)$$

4.2.1 Spoken sentence retrieval using single feature

This experiment used a total of 50 additional spoken keywords as input queries. The retrieval performance was determined for each individual feature. Table 4 shows the average frame numbers of l_q , l_d and $N_{\text{possible_seg}}$ determined by rectangular/Hamming window scanning/convolution.

Table 4. Average frame numbers of query, database sentence and possible segments.

Average l_q	70.64
Average l_d	345.23
Average $N_{\text{possible_seg}}$ by Rectangular window	37.72
Average $N_{\text{possible_seg}}$ by Hamming window	25.43

Table 5. Performance evaluation by single-feature.

Method	ASC	ASS	ASF	IHSC	IHSS	MFCC
Rectangular window	0.389	0.394	0.781	0.392	0.397	0.821
Hamming window	0.413	0.391	0.780	0.220	0.301	0.794
Direct matching	0.420	0.427	0.802	0.441	0.577	0.821

Table 5 presents the average precision by each feature. ASF had the best retrieval performance among all adopted MPEG-7 LLDs, while the retrieval performance of the other LLDs were below 0.43. The proposed methods degraded the mAP performance of the direct matching method by 0 ~ 0.033, except IHSC and IHSS.

4.3.2 Spoken sentence retrieval using multi-feature

Several combinations were considered: ASx (ASC + ASS + ASF), IHSx (IHSC + IHSS), LLDs (ASx + IHSx) and ALL (LLDs and MFCCs). Table 6 shows the average precision of multi-feature. The performance of the MPEG-7 LLDs combination was comparable to that of the MFCCs. Combining ASx with IHSx yielded no clear improvement. Moreover, the combination with MPEG-7 audio LLDs and MFCCs (MFCC + LLDs) gave the best retrieval performance.

Table 6. Performance evaluation by multi-feature.

Method	ASx	IHSx	LLDs	MFCC	MFCC+LLDs
Rectangular window	0.781	0.398	0.787	0.821	0.829
Hamming window	0.760	0.307	0.775	0.794	0.808
Direct matching	0.769	0.523	0.798	0.821	0.834

ASx: ASC + ASS + ASF; IHSx: IHSC + IHSS; LLDs: ASC + ASS + ASF + IHSC + IHSS; ALL: ASC + ASS + ASF + IHSC + IHSS + MFCC. The weights of ASC, ASS, ASF, IHSC, IHSS and MFCCs are 0.15, 0.15, 0.3, 0.08, 0.02 and 0.3, respectively.

Fig. 8 presents the overall precision-recall relations. The performance obtained from the multi-feature experiments was better than that obtained from the single-feature experiments. The performance, in terms of IHSx, Hamming window was poorer than that of rectangular window and the direct matching method. According to our observation, it was because the Hamming window poorly locates possible segments using IHSx. The curve of ASF was much more flat than that of other descriptors; its discrimination ability was much more outstanding. Finally, the loss of a few possible segments by the proposed approaches caused the curve to decline for better recall rate. However, as mentioned in section 3.2, the proposed algorithms greatly reduced the computational load and accelerate the retrieval process with only little precision degradation.

5. CONCLUSIONS

This paper presents a spoken sentence retrieval algorithm for resource limited devices. Rather than using a traditional large-vocabulary continuous-speech recognition system, it relies on a two-level feature matching technique. Without using acoustical and language models, the proposed system is language independent. Both MFCCs and MPEG-7 audio LLDs were also taken as speech features in our system evaluation. The retrieval performances of MPEG-7 audio LLDs were experimentally demonstrated to be comparable to the MFCCs feature. Furthermore, the combination of MPEG-7 LLDs and MFCCs performed better than either individually. The retrieval precision of the proposed methods were 1% ~ 3% lower than that of the direct matching method, however, the computational load was reduced by a factor of about l_q , the frames number of query.

REFERENCES

1. B. Chen, H. M. Wang, and L. S. Lee, "Discriminating capabilities of syllable-based

- features and approaches of utilizing them for voice retrieval of speech information in Mandarin Chinese,” *IEEE Transactions on Speech and Audio Processing*, Vol. 10, 2002, pp. 303-314.
2. H. M. Meng and P. Y. Hui, “Spoken document retrieval for the languages of Hong Kong,” in *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2001, pp. 201-204.
 3. S. E. Johnson, K. S. Jones, P. Jourlin, G. L. Moore, and P. C. Woodland, “The Cambridge university spoken document retrieval system,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, 1999, pp. 49-52.
 4. E. Chang, F. Seide, H. M. Meng, Z. Chen, Y. Shi, and Y. C. Li, “A system for spoken query information retrieval on mobile devices,” *IEEE Transactions on Speech and Audio Processing*, Vol. 10, 2002, pp. 531-541.
 5. K. Ng and V. W. Zue, “Phonetic recognition for spoken document retrieval,” in *Proceedings of the International Conference on Acoustics, Speech, Signal Processing*, 1998, pp. 325-328.
 6. M. Wechsler, “Spoken document retrieval based on phoneme recognition,” Ph.D. Dissertation, Swiss Federal Institute of Technology (ETH), Zurich, 1998.
 7. S. Srinivasan and P. Dragutin, “Phonetic confusion matrix based spoken document retrieval,” in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000, pp. 81-87.
 8. A. Singhal and F. Pereira, “Document expansion for speech retrieval,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 34-41.
 9. C. Fabio, “Towards the use of prosodic information for spoken document retrieval,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 420-421.
 10. 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., 2000.
 11. Y. Itoh, “A matching algorithm between arbitrary sections of two speech data sets for speech retrieval,” in *Proceedings of the International Conference on Acoustics, Speech, Signal Processing*, Vol. 1, 2001, pp. 593-596.
 12. O. Avaro and P. Salembier, “MPEG-7 systems: Overview,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, 2001, pp. 760-764.
 13. *Information Technology — Multimedia Content Description Interface — Part 4: Audio*, ISO/IEC CD 15938-4, 2001.
 14. B. Y. Ricardo and R. N. Berthier, *Modern Information Retrieval*, ACM Press, New York, 1999.
 15. W. K. Lo, H. Meng, and P. C. Ching, “Multi-Scale and multi-model integration for improved performance in Chinese spoken document retrieval,” in *Proceedings of International Conference on Spoken Language Processing*, 2002, pp. 1513-1516.
 16. J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, IEEE Press, New York, 2000.
 17. J. F. Wang, J. C. Wang, H. C. Chen, T. L. Chen, C. C. ChAng, and M. C. Shih, “Chip design of portable speech memopad suitable for persons with visual disabilities,”

- IEEE Transactions on Speech and Audio Processing*, Vol. 10, 2002, pp. 644-658.
18. L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, New Jersey, 1993.
 19. C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, 1980, pp. 623-635.
 20. E. M. Voorhees and D. K. Harman, "Appendix: Evaluation techniques and measures," in *Proceedings of the English Text Retrieval Conference*, 2000.
 21. MPEG-7 Requirements Group, "MPEG-7: Content and objectives," ISO/IEC JTC1/SC29/WG11, Sevilla, Spain, 1997.



Po-Chuan Lin (林博川) received the M.S. and Ph.D. degrees in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, in 2000, 2007, respectively. At present, he is engaged in the research and development of automatic minute/transcription generation system at Multimedia and Embedded System Design Laboratory, Department of Electronics Engineering and Computer Science, Tung Fang Institute of Technology. His research interests include speech signal processing, VLSI architecture design and embedded system design.



Jia-Ching Wang (王家慶) received the M.S. and Ph.D. degrees in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, in 1997, 2002, respectively. His research interests include signal processing and VLSI architecture design. Dr. Wang is an honor member of Phi Tau Phi. He is also a member of IEEE, ACM, IEICE, and International Speech Communication Association (ISCA).



Jhing-Fa Wang (王駿發) is now a Chair Professor in National Cheng Kung University, Tainan, Taiwan. He received his M.S. and B.S. degrees in the Department of Electrical Engineering from National Cheng Kung University, Taiwan in 1979 and 1973, respectively and Ph.D. degree in the Department of Computer Science and Electrical Engineering from Stevens Institute of Technology, U.S.A. in 1983. He was elected as an IEEE fellow in 1999 and now the chairman of IEEE Tainan Section. He got outstanding awards from Institute of Information Industry in 1991 and National Science Council of Taiwan in 1990, 1995, and

1997, respectively. He has developed a Mandarin speech recognition system called Venus-Dictate known as a pioneering system in Taiwan. He was an associate editor for IEEE Transaction on Neural Networks and VLSI System. He is currently leading a research group of different disciplines for the development of “Advanced ubiquitous media for created cyberspace”. He has published about 94 journal papers, 219 conference papers, and obtained 6 patents since 1983. His research areas include ubiquitous content-based media processing, speech recognition, and natural language understanding.



Jun-Jin Huang (黃俊憬) received the M.S. degree in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, in 2003. His research interests include speech signal processing and VLSI architecture design.