Robust DTW-based Recognition Algorithm for Hand-held Consumer Devices

Chanwoo Kim and Kwang-deok Seo, *Member*, IEEE

Abstract — *This paper presents a new robust Dynamic Programming (DP)-based recognition algorithm that is quite suitable for menu-driven recognition applications with small vocabulary size (typically less than 50). When compared to the conventional Dynamic Time Warping (DTW)-based recognizer, the proposed algorithm shows significantly improved recognition accuracy in speaker-independent cases. In addition, when compared to the conventional Hidden Markov Model (HMM)-based recognizer, the proposed algorithm requires smaller computational amount and parameter file size, while maintaining almost the same recognition rate for command recognition applications with small vocabulary size in hand-held consumer devices.*

Index Terms—**dynamic programming, speech recognition, speech processing.**

I. Introduction

Dynamic programming (DP) and its modifications have been successfully adopted for speech recognizer. This type of recognizer is commonly employed in handheld consumer devices like cell phones in the form of Dynamic Time Warping (DTW). Specifically, DTW-based recognition engine has been widely embedded inside Qualcomm MSM (mobile station modem) chips [1]. However, an inherent problem found in DTW algorithm is that it is vulnerable to speaker-independent (SI) recognition cases whereas it shows good performance for speaker-dependent (SD) cases [2], [10].

Although HMM-based recognition engine has begun to be employed for handheld devices due to its robustness in SI cases [10] and advantages in large vocabulary size and continuous speech recognition, DTW method still has various applicable areas including menu-driven commanding and phone dialing due to its low computational complexity and easiness in implementation. Moreover, in the case where the number of reference patterns is small, the required DB size needed for DTW method can also be kept small.

In this paper, we propose a robust DTW-based recognition algorithm that circumvents several inherent shortcomings of the conventional DTW algorithm. For this purpose, DP algorithm is employed in two levels: lower level and upper level. In the lower level, the algorithm is quite similar to the one that is used in the conventional DTW method while in the upper level we try to match the blocks of speech by using DP in order to obtain improved time alignment, thereby resulting in much higher recognition rate in the cases of long sentences and SI speech recognition. Even though an algorithm using two-level DP approach was already proposed for continuous speech recognition [3], our proposed algorithm is quite different from that one in that it obtains blocks of speech by partitioning each spoken sentence into voiced, unvoiced and silence parts and then performs DP matching for these blocks.

Recently, there have been several studies incorporating parameters related to voiced/unvoiced decision into HMM-based speech recognizer [7], [8], [9]. These parameters are employed either as a part of the feature vector [7], [9] or as a constraint in selecting an appropriate model [8]. As can be seen, these studies incorporate voiced/unvoiced decision resulting in improved recognition accuracy, while we exploit voiced/unvoiced and speech/silence decisions using DP in the proposed system.

Through extensive simulations using 500 recorded sentences and ETRI 611 DB [14], the proposed algorithm consistently shows improved recognition rate compared to the conventional DTW algorithm. The proposed algorithm shows not only improved robustness for SI speech recognition but also improved quality for inter-sex recognition.

The organization of this paper is as follows: In section II, previous works on the recognizer based on the template matching techniques are briefly reviewed and discussed. Section III introduces proposed robust DP-based recognition algorithm. Section IV includes several core experimental results to prove the robustness and superiority of the proposed method. The conclusions are provided in Section V.

II. Review of the Previous Works

Currently, there are largely two types of recognizers ported on the current cell phones. The first one is based on template matching technique [2]. The template matching technique has been widely used for cell phones for a long time. Recently, recognizers based on HMM have just become to be used in hand-held devices. This is due to the fact that they are more flexible in large vocabulary system, and shows better performance in SI cases. The ease of adding new vocabularies in the pronouncing dictionary is another advantage of HMM-based recognizers. However, recognizers based on the template matching techniques are still widely used for SD cases, since they are easy to implement, require small memory, and show impressive performance for SD cases. In this paper, we try to solve one inherent problem noticed in the template matching

C. W. Kim is with LG Electronics, Gasan-dong, Gumchon-gu, Seoul 153-081, Korea (e-mail: chanwcom@lge.com).
K. D. Seo is with Yonsei University, Wonju City, Gangwon 220-710, Korea (e-mail: kdseo@dragon.yonsei.ac.kr).

Manuscript received March 25, 2005

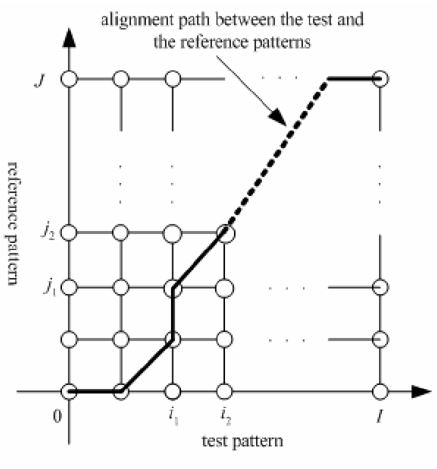


Fig. 1. Example of DTW time alignment.

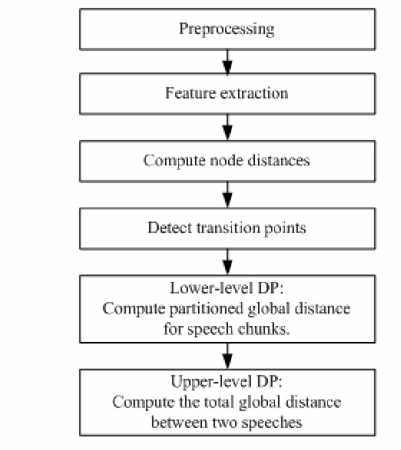

Preprocessing
Feature extraction
Compute node distances
Detect transition points
Lower-level DP:
Compute partitioned global distance for speech chunks.
Upper-level DP:
Compute the total global distance between two speeches

Fig. 2. Flow diagram of the proposed algorithm.

technique: poor performance for SI cases. Additional improvement can also be achieved such as robustness to inter-sex recognition.

In this section, we briefly review related conventional techniques for the proposed system. Template matching has long history in speech application [2]. Basically, it is based on the minimum distance between the test and the reference patterns along the aligned path, which is obtained using DP. In the conventional DP-matching technique, the plane of grids shown in Fig. 1 is generally utilized. This figure also shows an example of the alignment path. Global distance is the distance between the test and the reference patterns along the alignment path, and is computed along the alignment path. However, the alignment path is not apparent in the actual speech recognizers unless additional backtracking is performed.

The procedure for computing the global distance $\overline{D}(I,J)$ can be described by (1)-(3) [2], [17]:

i) Initialization

$$D(0,0) = d(0,0)m(0), \qquad (1)$$

ii) Recursion

For $0 \le i_2 \le I$, $0 \le j_2 \le J$, compute

$$D(i_2, j_2) = \min_{i_1, j_1}\{D(i_1, j_1) + d((i_1, j_1),(i_2, j_2))\}, \qquad (2)$$

where $d((i_1, j_1), (i_2, j_2))$ is the cost related to the movement from (i_1, j_1) to (i_2, j_2),

iii) Termination

$$\overline{D}(I,J) = \frac{D(I,J)}{\sum_{k=0}^{T} m(k)}. \qquad (3)$$

In the above equations from (1) to (3), a path from (i_1, j_1) to (i_2, j_2) is a single-step movement along the aligned path in Fig. 1. T in (3) is the total number of movement along the aligned path. The entire path starts from point of (0,0) and ends at (I,J). In (1) and (3), $m(k)$ denotes the weight associated with a movement. Studies on appropriate value of $m(k)$ can be found in [4], [23].

In the above procedure, time alignment between the test and the reference patterns is the most important factor. However, the alignment result might not be so satisfactory in the actual experiments. Moreover, when the alignment is unreliable, the recognition results tend to be quite inaccurate.

It is well known that the performance of a recognizer using template matching depends largely on the robustness of end-detection. Study by Wilpon et al. shows that the recognition accuracy degrades significantly, when the end-detection results are unreliable [11]. To alleviate this problem, Bridle suggested an algorithm to relax both of the end points and to find out the optimal points using DP [12]. By adopting this scheme, we can enhance the robustness against endpoint perturbation. Unlike the above procedure and conventional techniques in previous researches, we note that the following approaches are the most distinguished aspects of our paper to enhance the performance of recognizer.

i) In addition to the endpoints, we take the notable transition points inside both the reference and test patterns into account. The sequence of transition points in the test and the reference patterns might be quite different since there is a large variability in the uttered speech. However, using an additional DP algorithm to find the most matched transition points can resolve the problem.

ii) In this paper, the voiced/unvoiced decision is performed on each frame and the results are incorporated in obtaining the transition points. Speech/silence decision or voiced/unvoiced decision might not be always accurate. Additionally, the pattern of the transition points might differ for different speakers. Previous researches show that information on voiced/unvoiced decision can enhance the recognition accuracy[7]-[9]. However, the conventional recognizers based on template matching do not actually employ this feature in the real system. The additional DP used for the proposed system tries to find the best matched transition points and alleviates the problem. That is, by introducing additional DP, the recognizer can be more robust against the variability in the speech pattern and possible errors in speech/silence, voice/unvoiced decisions.

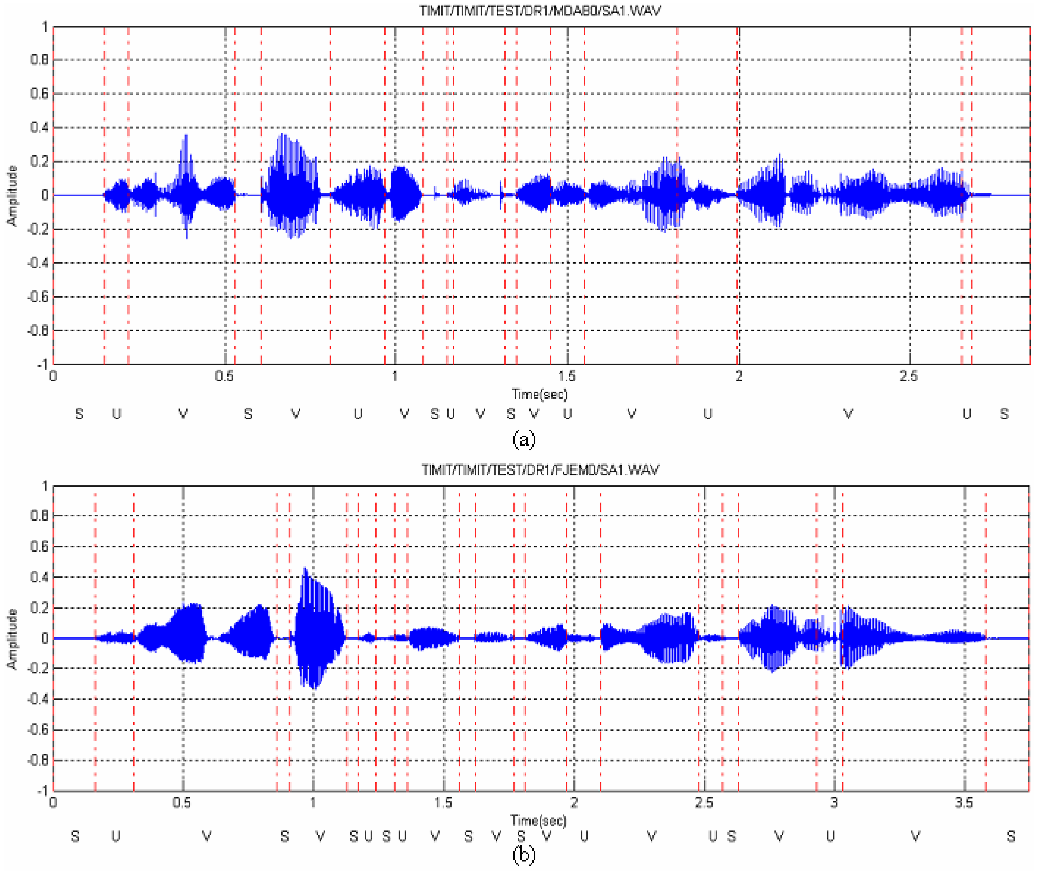


Fig. 3. Comparison of transition points in two speech samples in TIMIT DB: (a) TIMIT/TEST/DR1/MDAB0/SA1.WAV, (b) TIMIT/TEST/DR1/FJEM0/SA1.WAV.

III. Proposed Algorithm

The flow diagram of the proposed algorithm is illustrated in Fig. 2. In principle, we partition input speech into blocks of speech and apply DP in two levels. Partitioning is performed using notable transition points in the speech signal. Thereafter, in the lower level, partitioned global distance for each block is computed on a frame basis, which is quite similar to the case of conventional DTW algorithm. To find out the best-matched transition points, upper level DP is employed. By this upper level DP, we can obtain the total global distances between two speeches. Details on each stage are described in the following subsections.

Fig. 3 shows two speech samples in TIMIT DB [16], which are "*she had your dark suit in greasy wash water all year.*" spoken by two different speakers. Fig. 3(a) and (b) are TIMIT/TEST/DR1/MDAB0/SA1.WAV and TIMIT/TEST/DR1/FJEM0/SA1.WAV, respectively. In this figure, V, U, and S mean voiced, unvoiced, and silence blocks, respectively. As shown in this figure, there are many similarities between the characteristics of the blocks in speech samples although their patterns are not exactly the same. In the proposed study, we try to match the transition points defining these blocks by using a dynamic programming algorithm. Inside each block, the partitioned global distance is computed in a similar manner with the conventional DTW algorithm. However, the total global distance between these two speeches are obtained from the abovementioned additional DP. In the subsequent subsections, we will present these algorithms in detail. Thereafter, as an example, we will show the block alignment results for these two speech samples in Fig. 8.

A. Preprocessing

To enhance the accuracy of the detected transition points, we perform a simple amplitude normalization scheme. As will be explained, we use energy as a measure for detecting the transition points. It is well known that the relative intensity of the input speech significantly affects the performance of the speech/silence detection and voiced/unvoiced decision using energy. Thus, we normalize the intensity of the input speech to this recognizer. This procedure is performed in two steps. First, a Voice Activity Detector (VAD) finds the speech parts and computes the standard deviation value for these parts. Thereafter, based on the computed standard deviation of speech parts, we make this standard deviation a predefined constant by multiplying a gain. The predefined constant in the proposed system is fixed to be 1000 in the case of 16 bits-per-sample data. The accuracy of this constant is not so important.

TABLE I
CONSTRAINTS ON MATCHING TRANSITION POINT TYPES

	AE	VUV
Silence	$AE < 10^4$	All values
Voiced	$AE > 10^4$	$VUV > 0.4$
Unvoiced	$AE > 10^4$	$VUV < 0.4$

However, this processing is effective in that it makes the speech signals be unbiased by their relative intensities, which makes detecting the transition points of the test and reference speeches based on energy be more reliable.

B. Detecting Transition Point

As mentioned previously, we partition both the reference and test speeches into blocks. This procedure is performed based on the results of voiced/unvoiced decision and speech/silence decision on each frame. We perform the speech/silence decision based on the energy measure of the frame. The following equation shows the average energy for a single frame:

$$AE = \frac{1}{N_f} \sum_{n=0}^{N_f - 1} x^2[n] \tag{4}$$

where N_f is the frame length and x[n] is the speech signal in the frame [13]. In our experiment, the sampling rate of the speech signal is 8 kHz and the frame length is 10 ms. In this case, we find an energy threshold of 10^4 to be appropriate. Although the energy itself is not a sufficient measure for speech/silence measure, the preprocessing stage and the subsequent state machine scheme can enhance the detection rate.

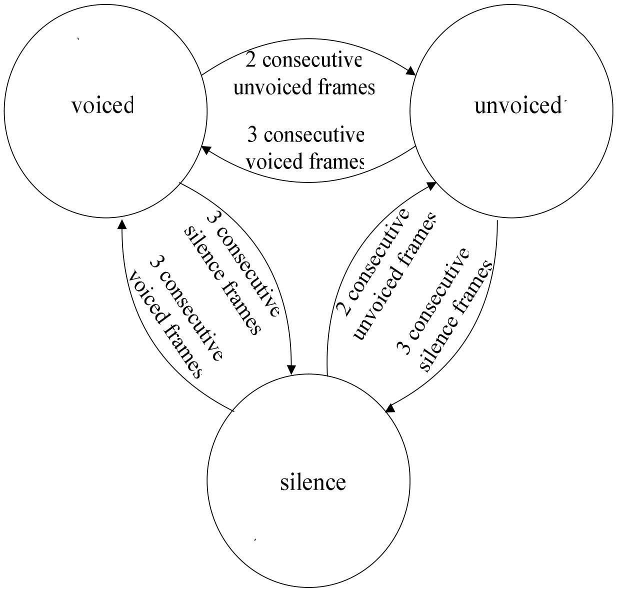

voiced
unvoiced
silence
2 consecutive unvoiced frames
3 consecutive voiced frames
3 consecutive silence frames
3 consecutive voiced frames
2 consecutive unvoiced frames
3 consecutive silence frames

Fig. 4. State transition diagram for detecting transition points.

In a similar manner, we perform the voiced/unvoiced decision using autocorrelation. This is based on the widely known fact that, for voiced parts, the periodicity is evident in the waveform, thus its autocorrelation function shows periodic peaks. We adopted the following measure for voiced/unvoiced decision:

$$VUV = \max_n \frac{r_x[n]}{r_x[0]}, \quad 16 \le n \le 160 \tag{5}$$

where r_x[n] is the autocorrelation function. Many previous

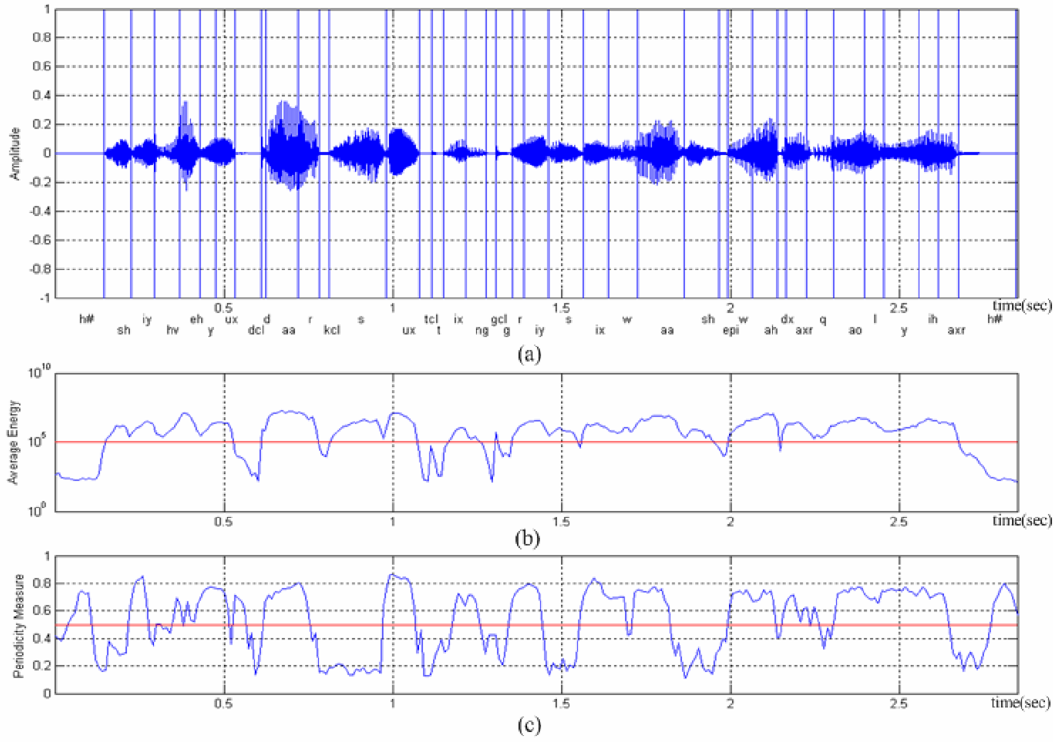


Fig. 5. (a) Phone locations, (b) Average energy, (c) Periodicity measure for a single speech sample in TIMIT DB (TIMIT/TEST/DR1/MDAB0/SA1.WAV).

researches show that *VUV* approaches 1 when the signal is quite periodic and this value is close to 0 when periodicity is almost inexistent in the signal [13], [17].

The range of *n* in (5) is for the case of 8 kHz sampling rate. This range is obtained from the fact that the pitch value lies between 50 *Hz* and 500 *Hz*. In the case of 16 *kHz* sampling rate as in TIMIT case, we should use $32 \leq n \leq 320$ instead. We adopted the threshold value of 0.4 by experiment. Many other researches also show that this value is suitable for voiced/unvoiced decision [17]. Table I summarizes the decision conditions for each frame.

While we decide whether a frame is silence, voiced speech, or unvoiced speech using above measures (4) and (5), some abrupt errors in decision may occur. To prevent these abrupt errors, we adopt a simple state transition scheme shown in Fig. 4. In this figure, there is a number on each arc. For a transition to occur, the measure obtained in (4) and (5) should satisfy the transition conditions at least for the specified number of frames. Specifically, for a transition from unvoiced to voiced state to occur, at least 3 consecutive frames should be decided to be voiced according to the measures (4) and (5).

We tested the accuracy of the above scheme for detecting transition points on the actual 500 speech samples. Each sample contains transition points ranging from 5 to 20. Compared to detection by human, this scheme yields accuracy over 90 %. While there still may be some small number of errors in detecting the transition points, it does not affect the overall performance of the recognizer significantly. This is due to the fact that the upper-level DP algorithm selects transition points that are best matched to the ones in another speech.

Since we classify each frame into voiced speech, unvoiced speech and silence by the above procedure, there are 6 possible transition types. Table II shows each transition type and the constraints on the possible matching for this type. We established this constraint not only to reduce the computational amount in the upper-level and lower-level DP, but also to prevent some unreasonable matching from ever happening. For example, silence-to-voiced transition in a reference pattern cannot be mapped onto voiced-to-silence transition in the test pattern, since they are quite different in their acoustical characteristics.

We perform the DP matching both in the lower and upper levels according to the constraints shown in Table II. In the lower level, we perform DP matching for a plane defined by speech blocks whose end points are transition points. If one of the end points is not allowed by this constraint, we do not have to perform the lower level DP matching.

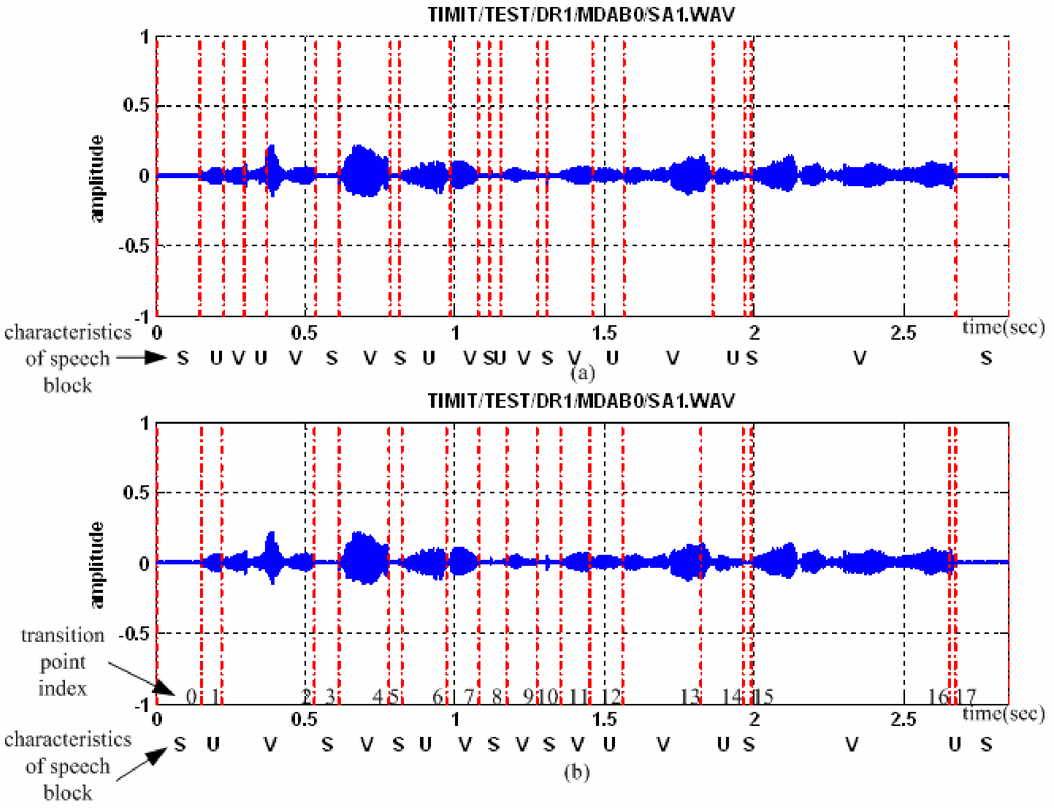


Fig. 6. Comparison of transition points in TIMIT DB:
(a) Obtained from labels in TIMIT DB, (b) Obtained using the algorithm adopted in the proposed recognizer.

TABLE II
CONSTRAINTS ON MATCHING TRANSITION POINT TYPES

Transition Type	Possible Matching Types
silence to voiced	silence to voiced silence to unvoiced
silence to unvoiced	silence to unvoiced silence to voiced
voiced to silence	voiced to silence unvoiced to silence
voiced to unvoiced	voiced to unvoiced voiced to silence
unvoiced to voiced	unvoiced to voiced silence to voiced
unvoiced to silence	unvoiced to silence voiced to silence

Similarly, in the upper level case, we perform DP matching only for grids satisfying this constraint in the same manner. The constraints in Table II are established by experiments considering acoustical characteristics.

Fig. 5 shows an example of phone locations, average energy, and voiced/unvoiced (VUV) measure for a sample speech in TIMIT. As widely known, TIMIT includes hand-labeled phone information. Fig. 5(a) shows the labels along with the speech waveform. Fig. 5(b) and (c) show the average energy *AE* and voiced/unvoiced decision measure *VUV*, respectively.

In Fig. 6, we compare the voiced/unvoiced decision results using the TIMT label files and the described procedure. In this figure, V, U, and S stand for voiced part, unvoiced part, and silence part respectively. The vertical dashed lines in this figure denote the transition point. Fig. 6(a) shows speech/silence and voiced/unvoiced decision results which are obtained from phone location information in TIMIT phone label file. Thus, the decision result in Fig. 6(a) is directly related to the phone location plot in Fig. 5(a). The decision results shown in Fig. 6(b) are obtained by the method described in this section. You can find that the location of transition points and decision results in Fig. 6(a) and Fig. 6(b) are very similar and therefore the method in this subsection works reliably. Using the TIMIT phone label file, we extensively evaluated the accuracy of the described method on TIMT DB. We obtained accuracy over 90 %, which is a quite similar result as our pervious experiment with 500 speech samples.

C. Lower-level DP Algorithm

In the proposed algorithm, after the transition points are obtained, we perform lower-level DP to compute the distance between blocks of test and reference patterns. The endpoints of each block are the previously obtained transition points.

Let us denote the transition points of the test and reference patterns by *I* and *J* respectively. If the number of transition points in the test and the reference patterns are I_{num_trans} and J_{num_trans} respectively, the ranges of *I* and *J* can be denoted by

$$0 \le I \le I_{num\_trans} \qquad (6)$$

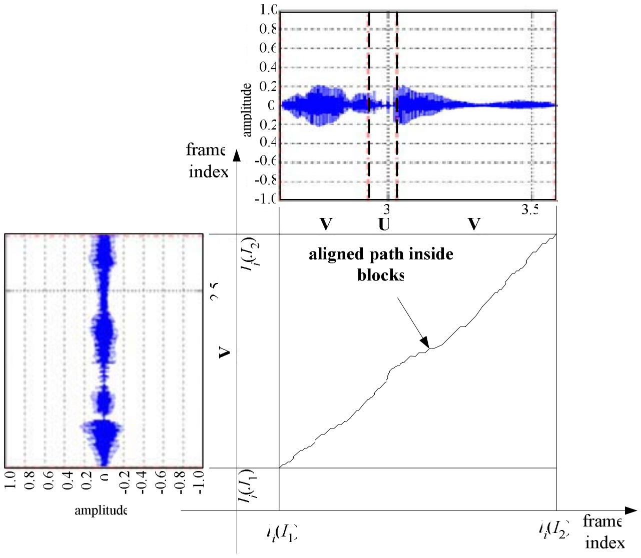


Fig. 7. Lower-level DP alignment.

$$0 \le J \le J_{num\_trans}. \qquad (7)$$

In Fig. 6(b), we show the transition point indices of a sample speech in TIMIT.

Fig. 7 illustrates an example of lower-level DP. Note that the speeches in the x-axis and y-axis are not the entire test or reference speeches but blocks obtained by the previous stage. As mentioned previously, the partitioned global distance, $P^{(I_1,J_1)\to(I_2,J_2)}$, is the distance between one block in the test speech and the other block in the reference speech. In this case, transition points I_1 and I_2 are the end points of a block in the test speech, and transition points J_1 and J_2 are the end points of a block in the reference speech. Then, the DP equations for obtaining $P^{(I_1,J_1)\to(I_2,J_2)}$ can be given as follows:

i) Initialization

$$P^{(I_1,J_1)\to(I_2,J_2)}(l_i(I_1),l_j(J_1)) = d(l_i(I_1),l_j(J_1))m(0)\,, \qquad (8)$$

ii) Recursion

For $l(I_1) \le i_2 \le l(I_2)$, $l(J_1) \le j_2 \le l(J_1)$, compute

$$P^{(I_1,J_1)\to(I_2,J_2)}(i_2,j_2) = \min_{i_1,j_1}\{P^{(I_1,J_1)\to(I_2,J_2)}(i_1,j_1) + d((i_1,j_1),(i_2,j_2)\} \qquad (9)$$

where $d((i_1, j_1), (i_2, j_2))$ is the cost related to the movement from (i_1, j_1) to (i_2, j_2),

iii) Termination

$$P^{(I_1,J_1)\to(I_2,J_2)} = P^{(I_1,J_1)\to(I_2,J_2)}(l_i(I_2),l_j(J_2)) \qquad (10)$$

$$m^{(I_1,J_1)\to(I_2,J_2)} = \sum_{k-0}^{T} m(k) \qquad (11)$$

where $m(k)$ is the path weight associated to k-th movement and $m^{(I_1,J_1)\to(I_2,J_2)}$ is the accumulated path weight.

In the above equations, $l_i(I)$ and $l_j(J)$ are the functions that relate a transition point index to a frame index in the test and reference speeches, respectively. Specifically, $l_i(I)$ is the frame index of I-th transition points in the test speech and $l_j(J)$ is the frame index of J-th transition points in the reference speech.

As you can see in the above equations, the procedure for lower-level DP algorithm is quite similar to the conventional DTW-based recognition algorithm which was described by (1) to (3). Typically, as shown in (3), conventional DTW algorithm performs the path weight normalization by dividing the accumulated distance by the sum of path weight [2], [17]. However, in (10), we do not divide the obtained accumulated distance $P^{(I_1,J_1)\to(I_2,J_2)}$ by the accumulated path weight $m^{(I_1,J_1)\to(I_2,J_2)}$. The reason is that the path weight normalization should be taken into account in subsequent upper-level DP that performs block alignment. For this reason, the accumulated path weight $m^{(I_1,J_1)\to(I_2,J_2)}$ as well as the partitioned global distance $P^{(I_1,J_1)\to(I_2,J_2)}$ should be stored for further use in the upper-level DP.

The above procedure described in (8) to (11) should be performed for all possible blocks. The constraints on end points I_1, I_2, J_1, and J_2 will be explained in the next subsection.

D. Upper-level DP Algorithm

After computing the partitioned global distances using (8)-(10), we compute the total global distance by matching the blocks. Fig. 8 shows an example of the actual alignment result for speech blocks in two TIMIT speech samples. This matching process is performed by an additional DP. For this stage, we use another plane of grids where each grid denotes the beginning and ending points of each speech block. In Fig. 8, you can see these grids. The resultant distance between the test and the reference patterns is the total weight.

The following equations (12) to (15) are employed in the upper-level DP stage:

i) Initialization

$$D(0,0) = 0\,, \tag{12}$$

ii) Recursion

For $0\leq I_2 \leq I_{num\_trans}$ and $0\leq J_2 \leq J_{num\_trans}$, compute

$$D(I_2, J_2) = \min_{I_1, J_1}\left\{D(I_1, J_1) + \alpha \cdot P^{(I_1,J_1)\to(I_2,J_2)}\right\} \tag{13}$$

where $P^{(I_1,I_2)\to(I_2,J_2)}$ is the cost related to the movement from (I_1, J_1) to (I_2, J_2),

iii) Termination

$$\overline{D}(I_{num\_trans}, J_{num\_trans}) = \frac{D(I_{num\_trans}, J_{num\_trans})}{\sum_{\phi} m^{(I_1,J_1)\to(I_{2,},J_2)}} \tag{14}$$

where $P^{(I_1,I_2)\to(I_2,J_2)}$ is the partitioned global distance obtained in (10). In (13) and (14), I_2 and J_2 are indices of I_2-th transition point of the test speech and J_2-th transition point of the reference speech, respectively. In the same way, I_1 and J_1 are I_1-th transition point of the test speech and J_1-th transition point of the reference speech, respectively. $D(I_2,J_2)$ means the sum of the partitioned global distances along the path from (0,0) to (I_2,J_2) that is obtained by DP. α in (13) is a weighing coefficient and will be explained in the next subsection E. In (14), as mentioned previously, the normalization is applied. ϕ denotes the path movement in this upper-level grid as shown in Fig. 8. $m^{(I_1,J_1)\to(I_2,J_2)}$ is the weight associated with the path movement and was obtained in (11).

In computation of this upper-level DP, we find that different local continuity constraint is necessary compared to the one adopted in the conventional DP-based recognizer. It is due to the fact that the DP matching in the upper level should be robust to the possible differences in the pattern of the transition points. According to our experiments, strict local continuity constraint sometimes results in performance degradation. You can also find the reason for this argument in Fig. 7 and Fig. 8. In these figures, the dotted lines partitioning speech waveforms denote transition points. As shown in these figures, the transition points are not the same in the test and the reference speeches. Note the solid line from $(l_i(I_1), l_j(J_1))$ to $(l_i(I_2), l_j(J_2))$ in Fig. 7 and the thick solid line for speech blocks from (I_1, J_1) to (I_2, J_2) in Fig. 8. In Fig. 7, two additional transition points are located inside a single block. In case of the thick solid line in Fig. 8, three additional transition points exist inside the block of the test speech while the aligned block of the reference speech does not contain any transition point. Thus, some of the transition points in one speech may not have corresponding transition points in the other speech. To take this fact into account, we need to adopt a less-constrained version of local path movement.

The following equation shows the local continuity constraint used in our system:

$$\begin{gathered} 1 \leq | I_2 - I_1 | \leq k \\ 1 \leq | J_2 - J_1 | \leq k \end{gathered} \tag{15}$$

where I and J are the transition point indices for test and reference patterns respectively as before. The adopted value of k in the proposed system is 4. Fig. 9 shows the recognition accuracy for various values of k.

The reason for adopting this constraint is that the patterns of transition points in the test and reference are not exactly the same as shown in Fig. 8. For example, the thick solid line in Fig. 8 shows the case of $|I_2 - I_1| = 4$ and $|J_2 - J_1| = 1$.

The local continuity constraints in (15) also constrain the possible values of I_1, I_2, J_1, and J_2. From the range of transition points given by (6) and (7), additional constraints on I_1, I_2, J_1, and J_2 are as follows:

$$\begin{gathered} 0 \leq I_1 < I_2 \leq I_{num\_trans} \\ 0 \leq J_1 < J_2 \leq J_{num\_trans.} \end{gathered} \tag{16}$$

Thus, we can conclude that (15) and (16) are constraints for the values of I_1, I_2, J_1, and J_2 in the lower and upper level DP described in (8) to (11) and (12) to (14).

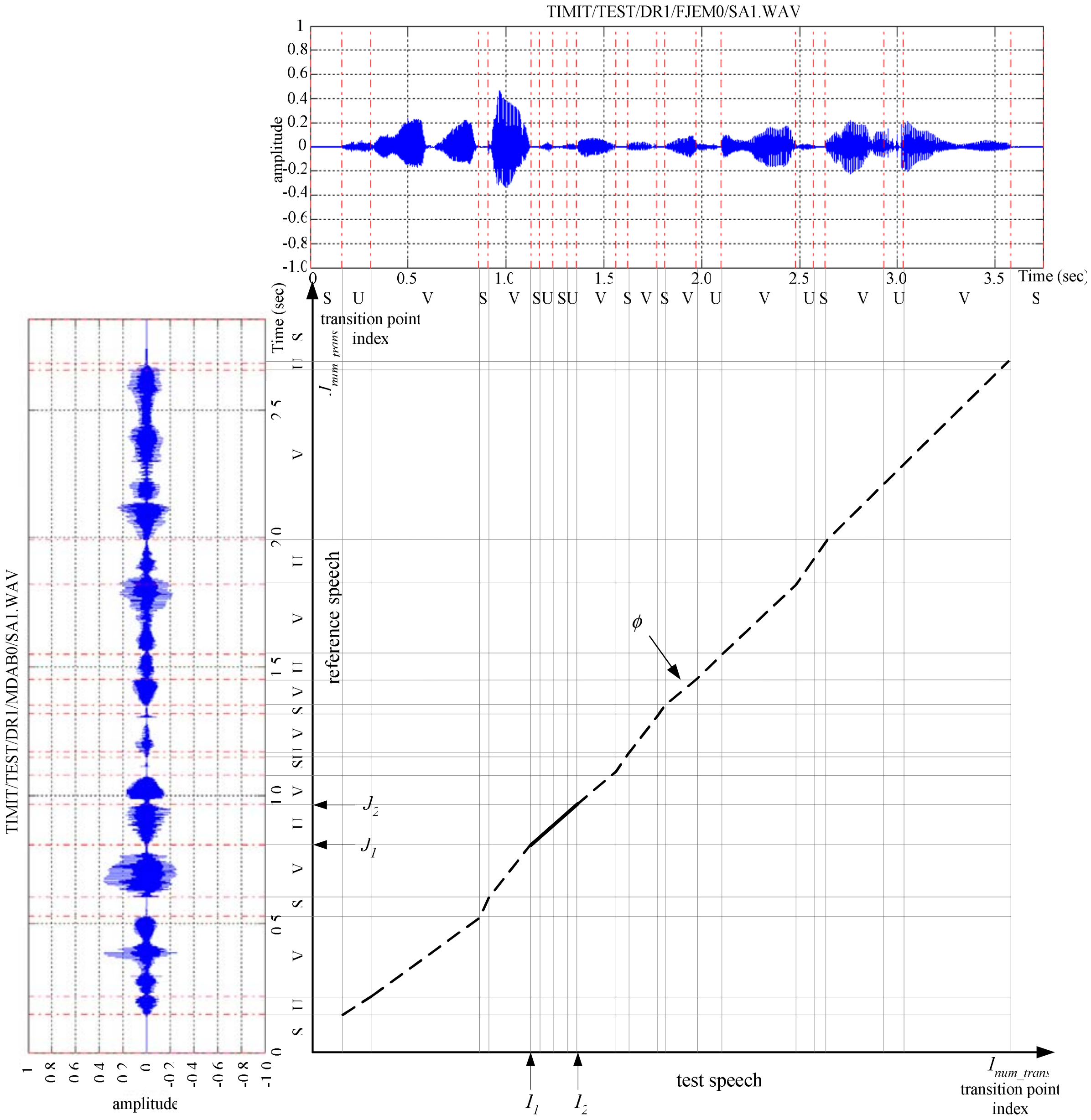


Fig. 8. The DP alignment in the upper level between two speech samples in TIMIT DB: TIMIT/TEST/DR1/FJME0/SA1.WAV and TIMIT/TEST/DR1/MDAB0/SA1.WAV.

E. Weighting on the Partitioned Speech

Proper alignment of the partitioned speech is crucial for the overall performance. In this subsection, we explain about a weighting coefficient α in (13) which is very important for the overall performance of the recognizer. It is quite natural that blocks of the same characteristics are more likely to be well-aligned than blocks of different characteristics. For example, as shown in Fig. 8, in most cases, voiced blocks in the test speech are aligned to voiced blocks in the reference speech. By introducing α, the alignment path is more likely to follow segments comprised of blocks of the same characteristics.

In the proposed system, α is 0.75 when the characteristics of blocks in the test and reference patterns are the same as in the case of voiced-voiced, unvoiced-unvoiced, and silence-silence. In other cases, α becomes 1. When different decision regions comprise a block, the most dominant decision in this block becomes the characteristic of the block.

Fig. 10 compares the accuracy of the recognizer for various weighting coefficients. The dashed horizontal line in this figure shows the recognition accuracy when the conventional recognition algorithm is used. The data in this figure were obtained by conducting experiments on 500 speech samples recorded using a condenser mic. In this experiment, 12th order MFCC was used for both the conventional and proposed

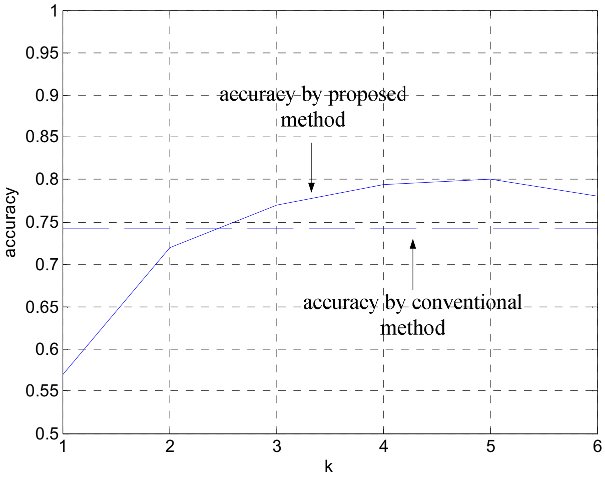


Fig. 9. Recognition accuracy for various values of *k*.

TABLE III

CONFIGURATION FOR THE BASELINE SYSTEM

Item	Value
Window type	Hamming window
Window length	30 ms
Feature type	12 th order MFCC (Zero order term included)
Frame period	10 ms
Preemphasis coefficient	0.95
Number of channels	26

algorithms. As can be seen in this figure, we note that a weighing coefficient somewhere between 0.7 and 0.75 is optimal. In the proposed recognizer, the weighting coefficient α of 0.75 is adopted.

F. Analysis on Computational Complexity

It has been generally believed by previous researches that computation of node distance constitutes around 80 % of the total DTW computational complexity [2]. And the rest computational load is taken up for DP matching. Note that the node distance computation part of the proposed system is exactly the same as that of the conventional ones.

In the lower level, the proposed algorithm requires only a little additional computation compared to the conventional DP matching. This is due to the fact that the lower level computation is basically the same as that of the conventional ones except that DP is applied for each partitioned blocks of speech. Moreover, the constraint described in Section III.B can even reduce the small additional computation. That constraint enables us not to compute all the partitioned global distances for each pair of speech blocks. Considering much smaller computational complexity for DP matching than that for node distance, the small additional computation required for DP matching in the lower level is quite ignorable.

It is found by simulations that typical spoken speech samples with duration of 1 to 3 seconds contain around 5 to 25 transition points. As explained before, upper level DP matching is performed on grids defined by these transition points. In the case of 10 ms frame length, there are 100 to 300 frames in the speech signal, which follows from the fact that 1s/10ms=100 and 3s/10ms=300. For the lower level DP matching, computation is done on each grid specified by frames. Thus, it is evident that the upper level DP requires much less computation than the lower level one. According to these analyses, we can conclude that there is no significant increase in the total computational amount for the proposed system compared to the conventional DTW algorithm.

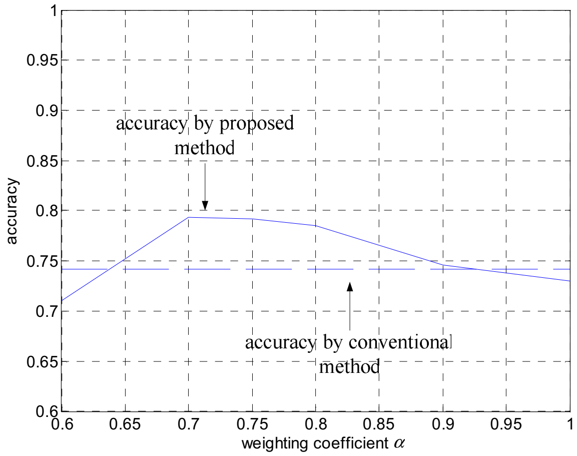


Fig. 10. Recognition accuracy for various weighting coefficients.

IV. EXPERIMENTAL RESULTS

To compare the proposed algorithm with a conventional DTW-based recognizer, the proposed algorithm is tested on 500 sentences spoken by 5 speakers. We recorded the test speech samples at 8 kHz sampling rate using a condenser microphone inside a hand-held device. Among 500 sentences, 100 speech samples are used for SD speech recognition cases and 400 samples are used for SI cases. In the test, the task of the recognizer is to select one of ten possible answers. In order to obtain various features in the evaluation phase, we used a feature extraction tool named *HCopy* that is a member of speech recognizer suite HTK [6].

For our baseline system, we use the configuration specified in Table III. Basically, this system is a recognizer based on 12th order MFCC (Mel Frequency Cepstral Coefficient).

For the above system, the recognition accuracy was 0.742. When the proposed algorithm was applied, we obtained an accuracy of 0.794, which corresponds to 5.2% gain over the original system.

In many previous researches, it is frequently reported that incorporating dynamic features like differential or acceleration coefficient enhances recognition accuracy [5]. We performed the tests using the proposed algorithm on these dynamic features. Table IV summarizes the recognition accuracy for the same speech samples when dynamic features are added to the feature vector. Previous researches have proved that cepstral

TABLE IV
RECOGNITION ACCURACY WHEN DYNAMIC FEATURES ARE INCLUDED

Feature Type	Conventional Algorithm	Proposed Algorithm
2th order MFCC	0.742	0.794
2th order MFCC (CMS applied)	0.784	0.804
12th order MFCC with differential coefficients	0.754	0.798
12th order MFCC with differential coefficients (CMS applied)	0.786	0.824
12th order MFCC with differential and acceleration coefficients	0.760	0.802
12th order MFCC with differential and acceleration coefficients (CMS applied)	0.792	0.828

TABLE V
RECOGNITION ACCURACY OBTAINED BY USING PLP

Feature Type	Conventional Algorithm	Proposed Algorithm
6th order PLP	0.812	0.830
6th order PLP (CMS applied)	0.832	0.846

TABLE VI
RECOGNITION ACCURACY FOR ETRI 611 DB SD CASES

Feature Type	Conventional Algorithm	Proposed Algorithm
6th order PLP	0.922	0.932
6th order PLP (CMS applied)	0.966	0.974

mean subtraction (CMS), as well as the dynamic features, is useful for robustness of a recognizer. In this table, we also include the simulation results when CMS scheme is employed.

Note that 2.0 to 5.4 % improvement in recognition accuracy was obtained by adopting the proposed algorithm. As shown in this table, you can also find that some improvements can be achieved by incorporating dynamic features. We also obtained notable performance improvement by adopting CMS.

We carried out experiments using PLP (Perceptual Linear Prediction) and the results are shown in Table V. In this experiment, we employed 6th order PLP coefficients and 10th order liftering coefficient. As shown in this table, a PLP feature vector of length 7 (6th order) shows better results than a MFCC feature vector with differential and acceleration coefficients whose length is 39 (12th order). Note that reducing the feature order results in small parameter file size. In the case of PLP, we could not obtain significant increase in recognition accuracy by employing dynamic features. The performance improvement obtained by employing PLP as a feature vector is generally larger than previous researches on HMM.

In SD cases, the improvement in speech recognition accuracy is not noticeable except for some of the long sentence cases, whereas the improvement reaches up to 5% for the SI cases. The insignificant improvement in SD cases is because the conventional DTW-based recognizer already shows good enough results. Thus little headroom is left over for further improvement. Considering the 5 speakers include a female, our system also shows robustness for inter-sex speech recognition cases. In the inter-sex recognition experiments on 80 speech samples, we obtained 3% increase in recognition accuracy.

For another comparison, we performed the same test using a triphone HMM-based recognizer using MFCC and obtained the recognition rate of 92% for the same test set, which is almost the same as that of the proposed system. While HMM-based recognizer needs a parameter file size of 748 KB, the proposed algorithm requires only 100 KB in this experiment. And for 10 vocabulary cases, the computational amount needed by the proposed algorithm is less than half of that is needed by the triphone HMM-based recognizer with pruning. We also find that for SD cases the DTW-based recognition engine shows better performance while HMM-based one shows superiority for SI cases. By adopting the proposed algorithm, we could reduce the differences in recognition accuracy for SI cases within 3%. According to this fact, the proposed algorithm is quite suitable for hand-held consumer devices with limited memory and CPU resources.

We perform another experiment using a monophone HMM-based recognizer with a single mixture. For our application, this recognizer requires almost the same parameter file size with the proposed system. However, for SD case, this simple HMM-based recognizer shows poor results, whose accuracy is lower than our system by more than 5 %. And for SI cases, the recognition accuracy is lower than the proposed system by 1 %. Note that the superiority of the proposed system over a HMM-based one only exists when the number of vocabulary is small. However, for applications like menu commanding in handsets, the proposed system shows advantageous aspects due to its low computational requirement and acceptable performance in accuracy.

In sum, the superiority of the proposed algorithm over the conventional DP-based recognizer is obvious. In some special applications with small vocabulary size and limited computational performance system, the proposed algorithm is advantageous compared to HMM-based algorithms.

Additionally, we conducted experiments on a standardized ETRI 611 database which comprises 611 Korean words spoken by 6 speakers. This speech corpus is widely used for training and testing speech recognizers for Korean words [14]. Using this DB, we performed experiments on 500 tasks. Each task is selecting the answer from 10 input speeches recorded in DB. Using 6th order PLP with CMS, we obtain the results in Table VI for SD cases. For SI case, we performed experiments on another 500 tasks. Table VII shows the results. As shown in this table, the proposed algorithm shows notable improvements in SI cases. When you compare this recognition results with

TABLE VII
RECOGNITION ACCURACY FOR ETRI 611 DB SI CASES

Feature Type	Conventional Algorithm	Proposed Algorithm
6th order PLP	0.870	0.898
6th order PLP (CMS applied)	0.912	0.930

the previous results on speech samples that are recorded with a condenser mic, you can find that much better result is obtained for ETRI 611 DB case. This can be explained by the fact that ETRI 611 DB is made with high fidelity mic in a noiseless environment. Moreover, ETRI 611 DB comprises relatively short words unlike the previous experiments using sentences.

Current CDMA handsets support DTW-based recognition by usually adopting Qualcomm proprietary solution [1] embedded inside MSM chip. We have now been continuing experiments on the proposed algorithm to have more competitive solution than the conventional DTW-based built-in solution and apply it to our commercial handsets under development.

V. CONCLUSIONS

In this paper, we propose a new robust DP-based speech recognition algorithm that is quite suitable for menu-driven recognition applications with small vocabulary size. The proposed algorithm shows improved robustness for SI cases compared to conventional DTW-based recognizers. This improvement can be explained in terms of DP matching using voiced/unvoiced and speech/silence information. By adopting this algorithm together with techniques like PLP and CMS, we could obtain even more improved results for the target application. The fact that it requires light computational complexity and small parameter file size makes it quite suitable for applications with small vocabulary size in hand-held consumer devices.

REFERENCES

[1] Qualcomm, *PureVoice^{TM} VR Toolkit Version 2.1.0 User Guide and Script Language Description*, San Diego, CA: Qualcomm, Mar. 2001.

[2] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[3] H. Sakoe, "Two-level DP matching – a dynamic programming-based pattern matching algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Process*ing, vol. ASSP 27, pp. 588-595, Dec. 1979.

[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP 26, pp. 43-49, Feb. 1978.

[5] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech, Signal processing*, vol. ASSP 34, pp. 52-59, Jan. 1986.

[6] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2)*. Cambridge, UK: Cambridge University Engineering Dept., 2002.

[7] D. O'Shaughnessy and H. Tolba, "Towards a robust/fast continuous speech recognition system using a voiced-unvoiced decision," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 413-416, Phoenix, AZ, Mar. 1999.

[8] D. Tomson and R. Chengalvarayan, "Use of periodicity and jitter as speech recognition features," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 24-28, Seattle, WA, May 1998.

[9] H. K. Kim and R. V. Cox, "A bitstream-based front-end for wireless speech recognition on IS-136 communication system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 9, no. 5, pp. 558-568, July 2001.

[10] C. Lévy, G. Linarès, and P. Nocera, "Comparison of several acoustic modeling techniques and decoding algorithms for embedded speech recognition systems," *Workshop on DSP in Mobile and Vehicular Systems*, Nagoya, Japan, Apr. 2003.

[11] J. G. Wilpon, L. R. Rabiner, and T. B. Martin, "An improved word-detection algorithm for telephone-quality speech incorporating both syntactic and semantic constraints, " *AT&T Tech. Journ*., vol. 63, no. 3, pp. 479-498, Mar. 1984.

[12] J. S. Bridle, and M. D. Brown, "Connected word recognition using whole word templates," in *Proc. of the Institute for Acoustics, Autumn Conference*, pp. 25-28, Nov. 1979.

[13] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice-Hall, 1978.

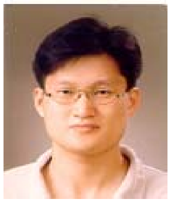
[14] J. Son, J. Kim, K. Kang and K. Bae, "Application of speech recognition with closed caption for content-based video segmentation," in *Proc. IEEE 9th DSP (DSP 2000) Workshop*, Hunt, TX, Oct. 2000.

[15] C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP 28, no. 6, pp.623-635, Dec. 1980.

[16] J. S. Garofolo, L. F.Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, *Darpa TIMIT Acoustic-Phonetic Continuous Speech Corpus,* Gaithersburg, MD: U. S. Department of Commerce, 1993.

[17] J. R. Deller, J. G. Proakis, J. H. L. Hansen, *Discrete-Time Processing of Speech Signals,* New York: Macmillan Publishing Company, 1993.

Chanwoo Kim received B.S. degree cum laude in electrical engineering and M.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 1998 and 2001, respectively. From 2000 to 2002, he worked on speech recognizers and embedded signal processing systems for Edumedia Technologies. Since 2003, he has been with LG Electronics. His research topics are multimedia and signal processing systems for handsets. His interests include speech recognition algorithm, speech analysis, multimedia systems, and embedded systems for signal processing.



Kwang-deok Seo received the B.S., M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1996, 1998, and 2002, respectively. From Aug. 2002 to Feb. 2005, he was with LG Electronics. Since March 2005, he has been a Faculty Member in the Computer and Telecommunication Engineering Division, Yonsei University, Wonju City, Korea, where he is an assistant professor. He received an Honor Prize and a Silver Prize at the Samsung HumanTech Thesis Prize in Feb. 2001 and Feb. 2002, respectively. His biographical profile has been included in the 7th and 8th Edition of *Marquis Who's Who in Science and Engineering®* and also in the 22nd Edition of *Marquis Who's Who in the World®*. He has over 30 pending or issued patents and has published over 30 papers in the areas of multimedia coding, multimedia signal processing, and multimedia communication systems. He is a member of KICS, IEEE and IEICE.