Efficient Speech Recognition System for Isolated Digits

Santosh V. Chapaneri[*1], Dr. Deepak J. Jayaswal[2]
[1]Assistant Professor, [2]Professor
Department of Electronics and Telecommunication Engineering
St. Francis Institute of Technology, University of Mumbai
Mumbai, India
[*]santoshchapaneri@gmail.com

Abstract— In this paper, an efficient speech recognition system is proposed for speaker-independent isolated digits (0 to 9). Using the Weighted MFCC (WMFCC), low computational overhead is achieved since only 13 weighted MFCC coefficients are used. In order to capture the trends of the extracted features, the local and global features are computed using the Improved Features for Dynamic Time Warping (IFDTW) algorithm. In this work, we propose to reduce the time complexity of the recognition system by time-scale modification using a SOLA-based technique and also by using a faster implementation of IFDTW. The experiments based on TI-Digits corpus demonstrate the effectiveness of proposed system giving higher recognition accuracy of 99.16% and performing about 22 times faster than conventional techniques.

Keywords-Speech Recognition; MFCC; Dynamic Time Warping, SOLA

I. INTRODUCTION

Automatic speech recognition (ASR) technology has made enormous advances in the last 20 years and has remained a popular area of research due to its importance as a natural man-machine interface. ASR is essentially a pattern recognition task, the goal is to take one pattern, i.e. the speech signal, and classify it as a sequence of previously learned patterns. An isolated-word speech recognition system requires that the speaker pause briefly between spoken words. With the recent commercial technology trends, speech recognition is often seen as an alternative to typing on a keyboard or touch/smart phones and tablets (eg. *Siri* from Apple and *S Voice* from Samsung).

The two main phases of ASR are training (feature extraction) and feature recognition. Several techniques for feature extraction exist including linear predictive coding (LPC) and Mel frequency cepstral coefficients (MFCC). LPC is a time-domain technique and suffers from variations in the amplitude of the speech signal due to noise [1]. The preferred technique is MFCC [2] since the cepstral features are compact, discriminable, and de-correlated. For feature recognition stage, several techniques are available including analysis methods based on Bayesian discrimination [3], Hidden Markov Models (HMM) [4], Dynamic Time Warping (DTW) based on dynamic programming [5], Support Vector Machines [6], Vector Quantization [7], and Neural Networks [8]. With the rapid development of intelligent signal processing and computational power, ASR has dominated its importance across multiple disciplines including heart sound diagnosis [9], recognition of digits spoken in different languages [10, 11], emotion recognition [12], recognition in hand-held consumer devices [13], and music retrieval [14]. However, most of these existing techniques suffer from a higher computational cost in the recognition stage because the time complexity of DTW increases as $O(N^2)$ as the number of samples in feature set increases. In [15], the author proposed modification of MFCC by using Weighted MFCC (WMFCC) which takes into account the original, delta and double-delta MFCC coefficients and combining them together to achieve a low-dimensional feature vector to reduce the processing time during recognition. Further in [15], the DTW algorithm is modified to include the local and global features of the WMFCC coefficients using Improved Features for DTW (IFDTW).

In this work, we propose to modify the work in [15] by reducing the processing time of overall system. Firstly the speech signal is compressed in time domain using a SOLA-based time scale modification technique while still preserving its perceptual characteristics and thus reducing the computational load on the recognition system. WMFCC is used for feature extraction stage to reduce the dimensionality of extracted features, and IFDTW technique is used for recognition stage. To reduce the time complexity of DTW, a fast IFDTW technique (FIFDTW) is proposed which speeds up the recognition stage. Experimental results demonstrate that we can speed up the processing by up to 22 times relative to existing techniques while still maintaining a good accuracy.

II. PRE-PROCESSING OF SPEECH SIGNAL

A. End Point Detection

To reduce the amount of processing, an end point detection algorithm [16] is applied to remove the silence and noise regions of speech signal while retaining the weak fricatives.

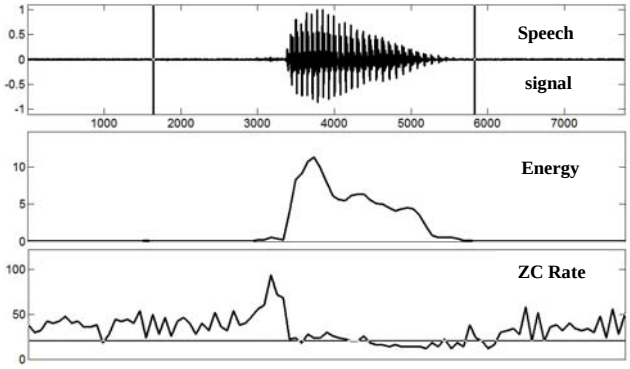

Figure 1. Result of end point detection for spoken digit "2"

The algorithm uses signal features based on energy levels and zero-crossing (ZC) rate. Fig. 1 shows the result of endpoint detection algorithm for spoken digit "2" along with its energy and zero-crossing rate plots. The solid vertical lines in the top plot indicate the portion of resultant speech void of silence regions but not excluding the weak fricative of '\t' from 'two'.

B. Time Scale Compression

Prior to feature extraction process in speech recognition system, we apply time scale compression on both the reference and test speech signals in order to reduce the duration of utterances. Time-scale modification (TSM) is often used to change the time scale of a signal. By trivial change of the playback rate, the speech is sped up twice but it sounds like a chipmunk effect. Preserving pitch and timbre of speech is important to maximize the intelligibility and quality of listening experience. While achieving pitch-scaling by first time-scaling and then resampling provides an efficient solution, it fails to preserve the frequency envelope structure [17], which can result in unnatural sound modifications being produced. A common technique to overcome this is to use Synchronized Overlap and Add (SOLA) algorithm [18], where the input (or analysis) signal $x[n]$ is segmented into overlapping frames of length N that are spaced S_a apart. The first frame is directly copied to the output (or synthesized) signal $y[n]$. Then, for any $m > 0$, the $(m+1)^{th}$ frame which starts at mS_a is shifted along the synthesized signal $y[n]$ around the target location mS_s within the range of $[k_{\min}, k_{\max}]$ to find a location that maximizes the cross-correlation function defined in (1), where L_k is the length of the overlapping region between the shifted frame and output signal. After finding the optimal location, the overlapping speech region is cross-faded and the rest of the analysis frame is copied to the synthesized signal. The time-scaling factor is obtained as $\alpha = S_s/S_a$.

$$R_{YX}[k] = \frac{\sum_{i=0}^{L_k-1} y[mS_s + k + i] \cdot x[mS_a + i]}{\sqrt{\sum_{i=0}^{L_k-1} y^2[mS_s + k + i] \cdot \sum_{i=0}^{L_k-1} x^2[mS_a + i]}} \quad (1)$$

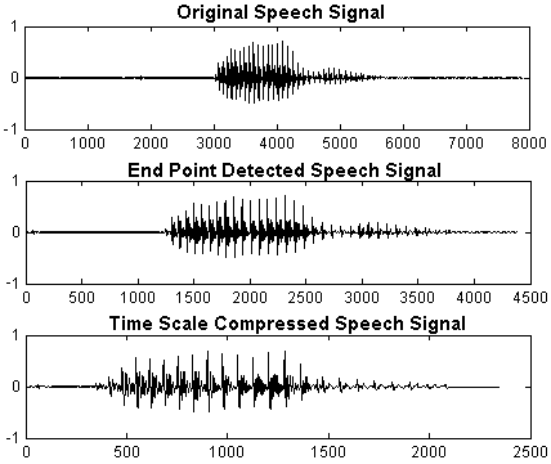

Figure 2. Time scale compression for spoken digit "7"

One major weakness of SOLA is the high computational cost required to search for the optimal overlapping point. In this work, Envelope Matching TSM (EM-TSM) [19] is used which significantly reduces the computational complexity of SOLA. It modifies the normalized cross-correlation function by using the sign information of the analysis and synthesized signals only. The modified function defined as envelope matching function is defined as (2) and (3). Fig. 2 shows the result of using EM-TSM on end-point detected spoken digit "7" with $\alpha = 0.5$, i.e. the speech signal is reduced to half time while still maintaining the pitch accurately.

$$R_{YX}[k] = \frac{\sum_{i=0}^{L_k-1} sign\{y[mS_s + k + i]\} \cdot sign\{x[mS_a + i]\}}{L} \tag{2}$$

$$sign(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{3}$$

III. Feature Extraction Module

The purpose of feature extraction is two-fold: first is to compress the speech signal into features, and second is to use features that are insensitive to speech variations, changes of environmental conditions and independent of the speaker. The speech spectrum is first pre-emphasized by approximately 20 dB per decade to flatten the spectrum of the speech signal [2]. Since the human speech signal is slowly time varying, it can be treated as a stationary process when considered under a short time duration [20]. Therefore, the speech signal is usually separated into frames where the typical frame length is 25 milliseconds and the neighboring frames are overlapped by 10 milliseconds. After segmenting into frames, each frame is multiplied by a Hamming window prior to the spectral analysis to reduce the discontinuity by attenuating the values of the speech samples at the beginning and end of each frame [21]. The spectral coefficients of each frame are computed using FFT which are then filtered using a Mel filter-bank of triangular bandpass filters [2]. The purpose of Mel filtering is to model the human auditory system that perceives sound in a nonlinear manner. The log Mel filter bank coefficients are computed from the filter-bank outputs as:

$$S(m) = 20\log_{10}\left(\sum_{k=0}^{N-1} |X(k)||H(k)|\right), \quad 0 < m < M \tag{4}$$

where M is the number of Mel filters, $X(k)$ is the N-point FFT of the frame, and $H(k)$ is the Mel filter transfer function [21]. The cepstrum is defined as the inverse Fourier transform of the log magnitude of Fourier transform of the signal. Since the log Mel filter bank coefficients are real and symmetric, the inverse Fourier transform operation can be replaced by DCT to generate the cepstral coefficients. This step is crucial in speech recognition as it can separate the vocal tract shape function from the excitation signal of the speech production model [20]. The cepstral coefficients are obtained as:

$$c(n) = \sum_{m=0}^{M-1} S(m)\cos\left(\pi n\left(m - 1/2\right)/M\right) \tag{5}$$

The first 13 cepstral coefficients are considered the MFCC features for each frame. The first cepstral coefficient is often replaced by log energy derived from the speech signal. To increase the accuracy of features as well as to reduce the dimensionality of features, liftering [22] and cepstral mean normalization is followed by the computation of the weighted MFCC features [15]. The trend of the speech signals in time is lost in the frame-

by-frame analysis. To recover the trend information, the time derivatives are computed using delta, double and triple delta features. The delta $\Delta c(n)$ features are calculated as follows:

$$\Delta c(n) = \frac{1}{\sum_{i=1}^{D} i^2} \sum_{i=1}^{D} i \times (c(n+i) - c(n-i)) \tag{6}$$

where $c(n)$ are the MFCC coefficients for each frame, and D is the frame delay, set to 2. Similarly, the double delta $\Delta\Delta c(n)$ and triple delta $\Delta\Delta\Delta c(n)$ features are calculated. These derived features are concatenated to the original cepstral features as shown in Fig. 3(a), thus giving a 52-dimensional MFCC feature vector for each frame. To reduce the dimensionality of features, a weighted MFCC (WMFCC) feature vector is used as follows:

$$wc(n) = c(n) + p \cdot \Delta c(n) + q \cdot \Delta\Delta c(n) + r \cdot \Delta\Delta\Delta c(n) \tag{7}$$

where the delta features are weighted according to p, q, and r. Since the derivative features contribute slightly less than $c(n)$, the weights are constrained to be $r < q < p < 1$. The feature vector $wc(n)$ is 13-dimensional thus reducing the complexity overhead of the recognition stage. Also as shown in Fig. 3(b), WMFCC and conventional MFCC have similar amplitude curves and can thus be effective in recognition as demonstrated in Section 5.

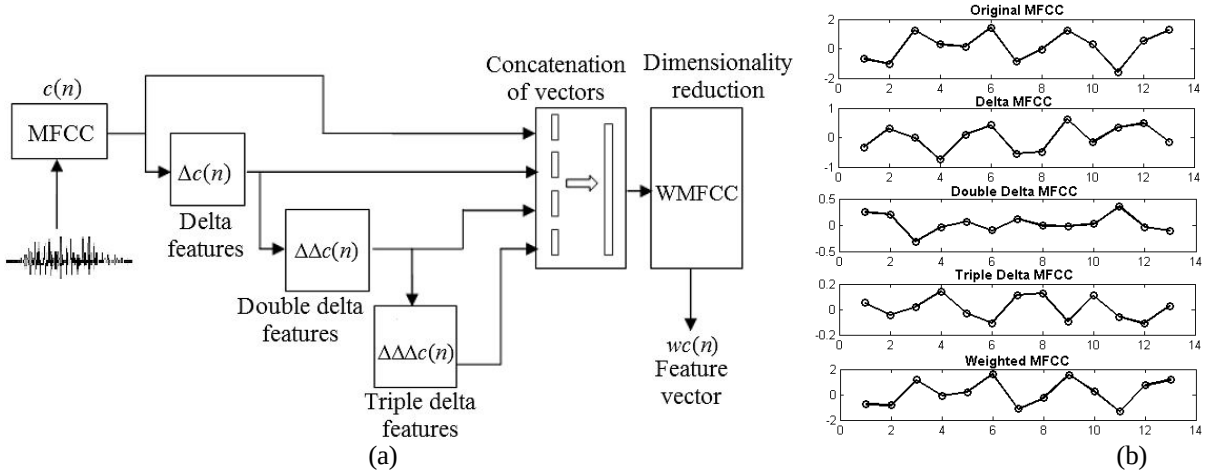

Figure 3. (a) Computation of WMFCC feature vector, (b) Feature vectors of a speech frame

IV. FEATURE RECOGNITION MODULE

A. *Conventional Dynamic Time Warping*

Dynamic Time Warping (DTW) is an algorithm adopted by the speech recognition community to handle the matching of non-linearly expanded or contracted signals [5]. Unlike Linear Time Warping (LTW) which compares two time series based on linear mapping of the two temporal dimensions, DTW allows a non-linear warping alignment of one signal to another by minimizing the distance between the two as shown in Fig. 4.

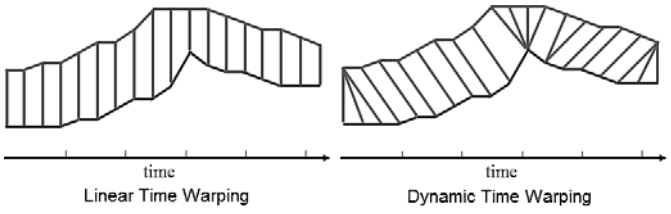

Figure 4. DTW non-linear alignment of two time series

The DTW algorithm finds the optimal path through a matrix of points representing possible time alignments between the signals. Based on Bellman's principle of optimality [21], the optimal alignment can be efficiently calculated via dynamic programming. Given two sequences $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$, the algorithm fills an m by n matrix representing the distances of best possible partial path using a recursive formula given by:

$$D(i, j) = d(i, j) + \min \begin{cases} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{cases} \tag{8}$$

where $1 \le i \le m$, $1 \le j \le n$, and $d(i, j)$ represents the distance between x_i and y_j. $D(1, 1)$ is initialized to $d(1, 1)$. The alignment that results in the minimum distance between the two sequences has the value $D(m, n)$. An example is shown in Fig. 5 for non-linear alignment of two similar sequences.

		P	A	T	T	T	E	E	R	R	R	N
	0	0	0	0	0	0	0	0	0	0	0	0
P	0	1	1	1	1	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	2	2	2	2
T	0	1	2	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
E	0	1	2	2	3	3	4	4	4	4	4	4
R	0	1	2	2	3	3	4	4	5	5	5	5
N	0	1	2	3	3	3	4	5	5	5	5	6

Figure 5. Minimum distance warping path for two time series

The warping path must satisfy the conditions of monotonicity, continuity, boundary and slope constraints [5]. There is also a constraint on adjustment window to speed up the calculations since an intuitive alignment path is unlikely to drift very far from the diagonal. The distance that the warp path is allowed to wander is limited to a band of size *R*, directly above and to the right of the diagonal. Fig. 6 illustrates the two window bands widely used in DTW computations.

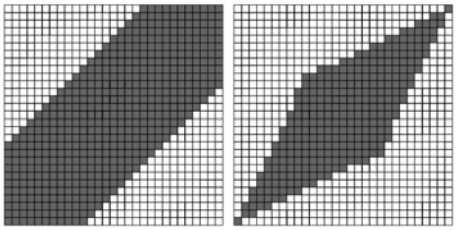

Figure 6. Adjustment window constraints: Sakoe-Chiba and (left) [5] and Itakura parallelogram with slopes of 2 & 0.5 (right) [23]

In application to speech recognition, the two time series corresponds to the two *numCoefficients* by *numFrames* WMFCC feature vectors of different speech signals. A two-dimensional cost matrix is computed that stores the minimum distance between two feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. The spoken digit's feature vector is compared to the template feature vectors using FIFDTW (discussed in Section IV.C) and the one with the minimum distance is chosen as recognition output.

B. DTW Modifications in Literature

The fundamental flaw of conventional DTW is that the numerical value of a data point in a time series does not represent the complete picture of the data point in relation to the entire sequence. In [24], derivative DTW was proposed in which each data point is replaced by its first derivative that serves as the local feature of a point expressing its relationship with two adjacent neighboring data points. The derivative estimates for MFCC feature vector *X* are computed as:

$$D(x_i) = \frac{(x_i - x_{i-1}) + (x_{i+1} - x_{i-1})/2}{2}, \quad 1 < i < K \tag{9}$$

where *K* is the number of frames. This estimate is not defined for the first and last vectors of the feature sequence. Similarly, derivative estimates are computed for the feature vector *Y* and conventional DTW algorithm is applied to these derivative features. Fig. 7 illustrates the alignment by the two techniques where we observe that the conventional DTW produces multiple singularities.

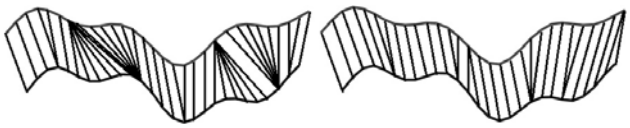

Figure 7. Alignment produced by conventional DTW (left) and derivative DTW (right) [adapted from 24]

In [25], quantized DTW (QDTW) algorithm was proposed where instead of storing multiple reference templates, it stores only one reference model for each spoken digit using vector quantization. For each spoken digit, the WMFCC vectors obtained in extraction module are first compared using the conventional DTW and

classes are created based on the optimal path. Centroids are calculated for each class and this vector is treated as the new reference. The process iterates till all the WMFCC vectors are compared and we obtain one reference vector for the corresponding spoken digit. However, the computational load is quite heavy for the vector quantization process in this algorithm thus slowing down the recognition performance.

In [26], sparse DTW algorithm is proposed which exploits the similarity between the input feature vectors (correlation) to find the optimal path. This allows using a sparse matrix to store the warping path instead of the full matrix.

In [15], improved features for dynamic time warping (IFDTW) algorithm was proposed (and used in this work) where instead of using absolute feature value or derivative estimates, modified features were used since an absolute value or local feature is not sufficient to identify and match common trends and patterns in the feature vectors. Both local and global features of each data point are used to track more accurately their contribution towards pattern matching. Fig. 8 illustrates the alignment between two feature vectors for the same digit spoken by two different speakers where the advantage of the IFDTW technique can be noted since it has fewer singularities compared to conventional DTW and derivative DTW [15]. However, the time complexity of IFDTW is same as that of conventional DTW and derivative DTW, i.e. $O(N^2)$. In this work, we propose to reduce the complexity by speeding up the IFDTW implementation.

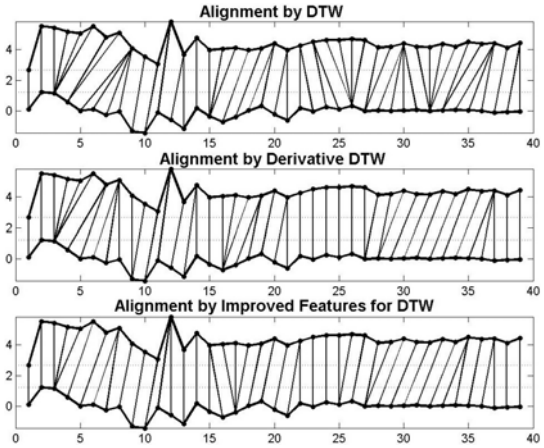

Figure 8. Alignment by various DTW techniques with minimum distance by DTW: 116, DDTW: 108, IFDTW: 94 [15]

C. Proposed FIFDTW Technique

To improve the time complexity of IFDTW, a fast IFDTW algorithm (FIFDTW) is implemeted which was proposed in [27]. Most of the techniques used to speed up DTW are based on three principles:

i) *Constraints* – limiting the number of cells for computation, eg. Sakoe-Chiba band [5] and Itakura constraints [23],

ii) *Data Abstraction* – performing DTW on a coarse representation of data [27], and

iii) *Indexing* – using lower bounding functions to reduce the number of times DTW has to run [28].

The concept of constraint has been already used in this work as shown in Fig. 6. When constraints are used, DTW finds an optimal warp path through the constraint window. However, the globally optimal warping path will not be found if it is not entirely inside the window. Using the constraints speeds up DTW by a constant factor, but IFDTW is still $O(N^2)$ if the size of the window is a function of the length of WMFCC feature vectors. In data abstraction, the DTW algorithm operates on a reduced representation of the data and the warping path becomes increasingly inaccurate as the level of abstraction increases. Directly projecting the low resolution warp path to the full resolution usually creates a warp path that is far away from the optimal, since it ignores local variations that could be significant. Indexing uses lower bounding functions to prune the number of times DTW algorithm is run, however it does not make the actual DTW calculation effiicent. FIFDTW is based on both constraints as well as abstraction approaches.

By data abstraction, FIFDTW first finds the optimal path through a coarse representation of data which is then refined to the original data set as illustrated in Fig. 9 using three stages:

i) *Coarsening* – shrinking the WMFCC feature vectors into smaller vectors that represent the same curve as accurately as possible with fewer data points,

ii) *Projection* – finding minimum distance warp path at a lower resolution, and using it as an initial guess for higher resolution's optimal path, and

iii) *Refinement* – refining the warp path by locally adjusting it using a radius parameter in the neighborhood of the projected path [27].

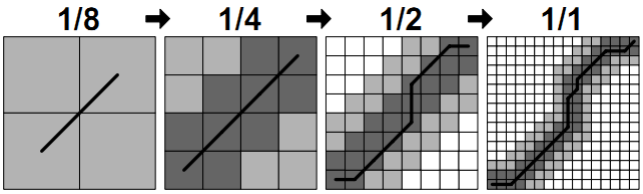

Figure 9. Refinement of optimal warping path in FIFDTW algorithm [adapted from 27]

It was demonstrated in [27] that fast DTW runs in O(*N*) time with sufficient accuracy, and thus this approach speeds up the proposed FIFDTW method significantly.

V. Experimental Results

The effectiveness of the proposed system using the efficient time-scale compression, WMFCC features and Fast Improved Features for DTW (FIFDTW) algorithms is tested for speaker-independent isolated spoken digits 0 to 9. The entire recognition system is implemented using Matlab. The training and test speech data are taken from TI-Digits database [29] from which samples from 15 male and 15 female speakers are used. Each digit is spoken twice by each speaker and total 600 utterances are collected. 360 utterances (60%) for training and 240 utterances (40%) are used for testing. In all experiments, the speech signal is divided into frames of duration 25 ms with 10 ms overlap between adjacent frames. The number of Mel filters used for feature extraction is 40 and 512-point FFT is used for WMFCC feature extraction stage.

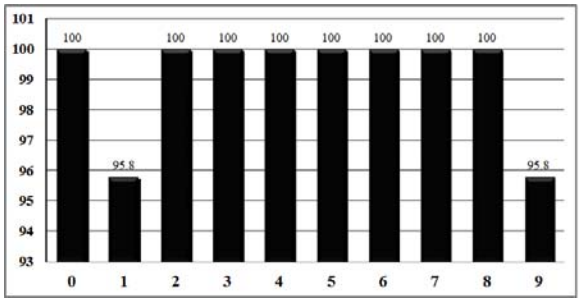

Figure 10. Recognition accuracy of proposed system for digits 0 to 9

Fig. 10 shows the recognition accuracy of the proposed speech recognition system for digits 0 to 9. Table I shows the confusion matrix for 24 test utterances of each digit from 0 to 9. The overall recognition accuracy of the proposed system is 99.16%.

TABLE I. Confusion Matrix

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	**24**	0	0	0	0	0	0	0	0	0
1	0	**23**	0	0	0	0	0	0	0	1
2	0	0	**24**	0	0	0	0	0	0	0
3	0	0	0	**24**	0	0	0	0	**0**	0
4	0	0	0	0	**24**	0	0	0	0	0
5	0	0	0	0	0	**24**	0	0	0	0
6	0	0	0	0	0	0	**24**	0	0	0
7	0	0	0	0	0	0	0	**24**	0	0
8	0	0	0	0	0	0	0	0	**24**	0
9	0	**1**	0	0	0	0	0	0	0	**23**

Table II compares the recognition accuracy obtained by various techniques, and Table III indicates the speed-up factors obtained by the proposed system at different time scale compression factors. At half-compression of speech signal, the accuracy obtained is 99.16% and the recognition is about 22 times faster. When the compression factor is too low, the recognition accuracy degrades because the EM-TSM (SOLA) algorithm distorts the input speech significantly. Hence it is proposed to use the time scale compression factor of no less than 0.5 to obtain the maximum recognition accuracy.

TABLE II. OVERALL RECOGNITION ACCURACY

	DTW	DDTW	FIFDTW
Conventional MFCC (13	85.97	88.57	91.63
Weighted MFCC (13 features)	93.38	96.75	**99.16**

TABLE III. SPEED-UP OF THE PROPOSED SYSTEM

Time Scale factor α	Accuracy (%)	CPU time (sec)	Speed-up factor
1	99.16	1.97	1
0.8	99.16	0.49	4.02
0.6	99.16	0.13	15.16
0.5	**99.16**	**0.087**	**22.64**
0.3	96.84	0.039	50.51
0.1	92.57	0.012	164.17
0.05	85.26	0.0081	243.21

VI. CONCLUSIONS

The proposed system is 22 times faster due to time-scale compression of the speech signal using a SOLA-based algorithm. For feature extraction, Weighted MFCC is used that considers both the voiceprint and the dynamic characteristics of the spoken digit, and Fast Improved Features for DTW (IFDTW) is proposed for feature recognition while reducing its time complexity using a faster implementation of DTW. By enhancing the WMFCC features considering their local and global trend over the entire spoken speech signal in IFDTW, higher accuracy is achieved compared to using the conventional DTW (pure value based) and derivative DTW (local feature based) algorithms. The experimental results demonstrate that the proposed system with WMFCC and FIFDTW achieves higher accuracy and has a lower computational overhead on the feature extraction as well as the recognition stage.

REFERENCES

[1] J. Tierney, "A study of LPC analysis of speech in additive noise", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 389-397, Aug 1980

[2] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993

[3] V. Mitra, N. Hosung, C. Wilson, E. Saltzman, and L. Goldstein, "Gesture-based dynamic bayesian network for noise robust speech recognition", *IEEE Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5172-5175, May 2011

[4] C. Do, D. Pastor, and A. Goalic, "On the recognition of cochlear implant-like spectrally reduced speech with MFCC and HMM-based ASR", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1065-1068, July 2010

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

[6] W. Tarng, Y. Chen, K. Hsieh, K. Ou, C. Li, and M. Chen, "Applications of support vector machines on smart phone systems for emotional speech recognition", *Intl. Journal of Research and Reviews in Computer Science*, vol. 1, no. 4, Dec 2010

[7] S. Singh, and E. Rajan, "Vector Quantization approach for speaker recognition using MFCC and inverted MFCC", *International Journal of Computer Applications*, vol. 17, no. 1, Mar 2011

[8] G. Dede, and M. Sazli, "Speech recognition with artificial neural networks", *Digital Signal Processing (Elsevier)*, vol. 20, no. 3, pp. 763-768, May 2010

[9] W. Fu, X. Yang, and Y. Wang, "Heart sound diagnosis based on DTW and MFCC", *3rd IEEE Intl. Congress on Image and Signal Processing*, vol. 6, pp. 2920-2923, Oct 2010

[10] S. Ghanty, S. Shaikh, and N. Chaki, "On recognition of spoken Bengali numerals", *IEEE Intl. Conf. Computer Information Systems and Industrial Management Applications*, pp. 54-59, Oct 2010

[11] A. Mishra, M. Chandra, A. Biswas, and S. Sharan, "Robust features for connected Hindi digits recognition", *Intl Journal of Signal Processing, Image Processing, and Pattern Recognition*, vol.4, no. 2, pp. 79-90, June 2011

[12] N. Sato, and Y. Obuchi, "Emotion recognition using Mel-frequency cepstral coefficients", *Journal of Natural Language Processing*, vol. 14, no. 4, pp. 83-96, Sep 2007

[13] C. Kim, and K. Seo, "Robust DTW-based recognition algorithm for hand-held consumer devices", *IEEE Trans. Consumer Electronics*, vol. 51, no. 2, pp. 699-709, May 2005

[14] Z. Guo, Q. Wang, G. Liu, J. Guo, and Y. Lu, "A music retrieval system using melody and lyric", *IEEE Intl. Conf. Multimedia and Expo Workshops*, pp. 343-348, July 2012

[15] S. Chapaneri, "Spoken digits recognition using weighted MFCC and improved features for dynamic time warping", *International Journal of Computer Applications*, Vol. 40, No. 3, pp. 6-12, Feb 2012

[16] L. Rabiner, and M. Sambur, "An algorithm for determining the endpoints of isolated utterances", *Bell System Technical Journal*, vol. 54, no. 2, pp. 297-315, Feb 1975

[17] Bristow-Johnson, R., "A detailed analysis of a time-domain formant-corrected pitch-shifting algorithm", *Journal of the Audio Engineering Society*, Vol. 43, No. 5, pp. 340-352, May 1995

[18] S. Roucos, and A. Wilgus, "High quality time scale modification for speech", *Proc. IEEE Intl. Conf. Acoustics, Speech, Signal Processing*, Vol. 1, pp. 493-496, Apr 1985

[19] J. Wong, O. Au, and P. Wong, "Fast time scale modification using envelope-matching technique (EM-TSM)", *Proc. IEEE Intl Symp. Circuits and Systems*, Vol. 5, pp. 550-553, May 1998

[20] J. Picone, "Signal modeling techniques in speech recognition", *Proc. of the IEEE*, vol. 81, no. 9, Sep 1993

[21] J. Deller, J. Proakis, and J. Hansen, *Discrete Time Processing of Speech Signals*, Prentice Hall, NJ, USA, 1993

[22] B. Juang, L. Rabiner, and J. Wilpon, "On the use of bandpass liftering in speech recognition", *IEEE Intl. Conf. Acoustics, Speech, and Signal Processing*, pp. 765-768, Apr 1986

[23] F. Itakura, "Minimum prediction residual principle applied to speech recognition", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-23, pp. 52-72, Feb 1975

[24] E. Keogh, and M. Pazzani, "Derivative dynamic time warping", *Proc. of the 1st SIAM Intl. Conf. Data Mining*, Chicago, USA, 2001

[25] T. Zaharia, S. Segarceanu, M. Cotescu, A. Spataru, "Quantized dynamic time warping (DTW) algorithm," *IEEE Intl. Conf. Communications*, pp.91-94, June 2010
[26] Al-Naymat, Ghazi, Sanjay Chawla, and Javid Taheri. "SparseDTW: A novel approach to speed up dynamic time warping." *Proc. 8th Australian Data Mining Conference*, Vol. 101, pp. 117-127, 2009
[27] S. Salvador, and P. Chan, "FastDTW: toward accurate dynamic time warping in linear time and space", *Proc. of 3rd KDD Workshop on Mining Temporal and Sequential Data*, pp. 70-80, Aug 2004
[28] D. Lemire, "Faster retrieval with a twopass dynamic-time-warping lower bound", *Pattern Recognition*, Vol. 42(9), 2169–2180, 2009
[29] R. Leonard, "A database for speaker-independent digit recognition", *IEEE Intl. Conf. Acoustics, Speech, and Signal Processing*, pp. 328-331, Mar 1984