S13.8

A TMS32020-BASED REAL TIME, TEXT-INDEPENDENT, AUTOMATIC SPEAKER VERIFICATION SYSTEM

Joseph B. Attili

PAR Technology Corp.
220 Seneca Turnpike
New Hartford, New York 13413

Michael Savic

Electrical, Computer, and Systems Engineering Dept.
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

Joseph P. Campbell, Jr.

U. S. Department of Defense
Fort Meade, Maryland 20755-6000

ABSTRACT

This paper describes a fast, reliable, yet inexpensive automatic speaker verification system based around the Texas Instruments TMS32020 digital signal processor (DSP). This system employs a novel speaker verification algorithm which operates in seventy-five percent of real time and requires two to three seconds of unconstrained speech to perform accurate authentication. Experimental results on a population of ninety speakers are also given.

1. INTRODUCTION

Automatic speaker verification (ASV) is a process by which a machine authenticates the claimed identity of a person from his or her voice characteristics. A machine capable of such a task would be an invaluable tool for numerous security and forensic applications. This paper describes one such system developed at Rensselaer Polytechnic Institute [1].*

All the signal processing for the ASV system developed in this research is performed on a single custom-designed board based around a Texas Instruments TMS32020 fast digital signal processor. This system is a fast, reliable, yet inexpensive unit which operates in approximately seventy-five percent of real time and requires two to three seconds of unconstrained speech to perform accurate verification. (An IBM PC serves as the user interface for the system.)

A novel speaker verification algorithm was implemented on the TMS32020 hardware. Instead of using a single set of features (e.g., LPC coefficients), several families of features are combined to obtain better recognition accuracy. This, however, significantly increases the computational complexity of the pattern recognition problem. In order to allow the speaker verification algorithm to operate in real time, a new feature selection criterion was developed and implemented which selects *speaker-dependent* features. That is, the features which are used for classification are different for each of the users of the system.

Several classification schemes were examined for application in this system. The rule implemented in the final prototype uses a sequential strategy to arrive at the authentication decision. Hence, the users of the system are required to speak until a decision with high confidence is announced.

The paper is composed as follows. Section 2 presents a review of the speaker verification algorithm. Section 3 discusses the hardware implementation. Finally, section 4 summarizes some experimental testing of the algorithm on a population of ninety speakers.

2. A NOVEL SPEAKER VERIFICATION ALGORITHM

The speaker verification algorithm developed in this research is divided into two tasks - training and verification. Common to both phases are the data acquisition and preprocessing of the speech signal. In particular, the analog waveform is lowpass filtered to 4 KHz, digitized to 12 bits at a rate of 10 KHz, spectrally conditioned using a 95% preemphasized filter, and Hamming windowed into nonoverlapping 20 ms analysis frames. Silence removal is also performed by a simple energy threshold.

The next step in either phase is "raw" feature extraction on those frames judged to be speech. This involves a standard 12^{th}-order autocorrelation LPC analysis. The resulting PARCOR and LPC coefficients are then transformed into cepstrum and log-area coefficients. A normalized gain feature is also extracted.

One novel attribute of the speaker verification is the fact that several of these nonlinearly related feature sets are combined together to form an augmented feature space. In particular, an N=37 dimensional feature vector consisting of

- twelve PARCOR coefficients,
- twelve log-area coefficients,
- twelve LPC cepstrum coefficients,
- and one normalized gain coefficient

is composed to represent the speech for each frame. Since this enlargement of the feature space greatly increases the dimensionality of the pattern recognition problem, and hence its computational complexity, a new feature selection method was developed to reduce the size of the feature space. This technique can be summarized as follows.

2.1 Training

Let $\underline{x}_n$ represent a feature vector in the original space corresponding to the n^{th} analysis frame. When training the ASV system, the mean, $\underline{\mu}_i$, and covariance, $\underline{\underline{K}}_i$, of the $\underline{x}_n$'s for each of the c users of the system, $i=1,2,..,c,$ are estimated from training data:

$$\underline{\mu}_i = \frac{1}{NF_i} \sum_{n=1}^{NF_i} \underline{x}_n, \tag{1}$$

$$\underline{\underline{K}}_i = \frac{1}{NF_i} \sum_{n=1}^{NF_i} [\underline{x}_n-\underline{\mu}_i][\underline{x}_n-\underline{\mu}_i]^T, \tag{2}$$

* This work was supported by the U. S. Department of Defense.

where NF_i is the number of training vectors available for speaker i. The overall population *within-class scatter*, $\underline{\underline{S}}_w$, and the *between-class scatter* for speaker i, $\underline{\underline{S}}_{bi}$, are then computed according to

$$\underline{\underline{S}}_{w} = \sum_{m=1}^{c} \underline{\underline{K}}_{m}, \tag{3}$$

$$\underline{\underline{S}}_{bi} = \sum_{m=1}^{c} [\underline{\mu}_m - \underline{\mu}_i][\underline{\mu}_m - \underline{\mu}_i]^T. \tag{4}$$

An MxN ($M<<N$) *feature selection matrix*, $\underline{\underline{A}}_i$, is then determined which optimizes the following trace criterion:

$$J = \mathrm{tr}\left\{ \underline{\underline{S}}'^{-1}_{w} \; \underline{\underline{S}}'_{bi} \right\}, \tag{5}$$

where

$$\underline{\underline{S}}'_{w} = \underline{\underline{A}}_i \underline{\underline{S}}_w \underline{\underline{A}}_i^T, \tag{6}$$

and

$$\underline{\underline{S}}'_{bi} = \underline{\underline{A}}_i \underline{\underline{S}}_{bi} \underline{\underline{A}}_i^T \tag{7}$$

are the scatter matrices of the training data when projected onto the M-dimensional subspace chosen for speaker i. (It is well known that the optimal solution for $\underline{\underline{A}}_i$ is the matrix of eigenvectors corresponding to the M-largest eigenvalues of $\underline{\underline{S}}_w^{-1}\underline{\underline{S}}_{bi}$.) Figure 1 is a typical scatter plot showing the projections of the feature data for several speakers projected onto the selected subspace (two-dimensional) for speaker #1.

The important difference between this feature selection technique and the traditional multiclass discriminant analysis is in the definition of the between-class scatter, (4). In the common multiclass discriminant analysis procedure [2], $\underline{\underline{S}}_b$ is formed so that the normalized distances of the projected means for each class from the overall population mean are as large as possible; e.g.:

$$\underline{\underline{S}}_{b} = \sum_{m=1}^{c} [\underline{\mu}_m - \bar{\underline{\mu}}][\underline{\mu}_m - \bar{\underline{\mu}}]^T, \tag{8}$$

where

$$\bar{\underline{\mu}} = \frac{1}{c} \sum_{m=1}^{c} \underline{\mu}_m. \tag{9}$$

Thus, the features selected are the same for all classes. In contrast, our method selects features so that the projected clusters of all known speakers are moved as far as possible from that of the speaker to be verified. Thus, the features selected for a particular subject are indicative of that particular speaker and, hence, the features used for authentication are *speaker dependent.*

2.2 Verification

When verifying the i^{th} talker, the N-dimensional "raw" feature data for the n^{th} analysis frame of the verification utterance is projected onto the much lower dimensional subspace specified by $\underline{\underline{A}}_i$:

$$\underline{y}_n = \underline{\underline{A}}_i \underline{x}_n. \tag{10}$$

The projected data, $\underline{y}_n$, is modelled as being distributed as an M-dimensional Gaussian variates with mean $\underline{\underline{A}}_i \underline{\mu}_i$ and covariance matrix $\underline{\underline{A}}_i \underline{\underline{K}}_i \underline{\underline{A}}_i^T$ We further assume that $\underline{y}_m$ and $\underline{y}_n$ are independent and identically distributed for m≠n. Under these assumptions, it can be shown [1] that the log-likelihood ratio Bayes-optimal decision rule reduces to

$$l\{\underline{y}\} = \mathrm{tr}\{ \underline{\underline{Q}}_i \bar{\underline{\underline{R}}}_{NF} \} - 2\bar{\underline{y}}_{NF}^T \underline{z}_i, \tag{11}$$

where

$$\bar{\underline{y}}_{NF} = \sum_{k=1}^{NF} \underline{y}_k, \tag{12}$$

$$\bar{\underline{\underline{R}}}_{NF} = \sum_{k=1}^{NF} \underline{y}_k \underline{y}_k^T \tag{13}$$

are first and second order statistics over the verification utterance, $\underline{\underline{Q}}_i$ and $\underline{z}_i$ are parameters computed from training data, and NF is the number of frames observed thus far in the authentication utterance. (In this decision rule, the "universe" of speakers is divided into two classes - speaker i and the rest of the user population enrolled onto the system.)

Instead of basing the authentication decision on a fixed number of feature vectors (analysis frames) and a single threshold, our system uses a speaker dependent sequential decision strategy. That is,

if $l\{\underline{y}\} > \Lambda_{Ai}$, then customer is *accept*ed as speaker i;

if $l\{\underline{y}\} < \Lambda_{Bi}$, then customer is *reject*ed as speaker i;

Otherwise, another frame is requested.

The thresholds Λ_{Ai} and Λ_{Bi} are chosen (automatically) as a compromise between three factors - the type 1 and type 2 error rates and the average number of frames required for verification.

As a matter of practical concern, a minimum number of frames must be input before a verification decision is attempted. Moreover, if no decision is reached after some maximum number of frames, the speaker is automatically rejected. (Figure 2 shows a flowchart of the verification phase of the algorithm.)

In summary, the speaker verification algorithm developed in this research uses speaker-dependent features which are derived from a combination of several families of features. Furthermore, a sequential decision strategy with speaker-dependent parameters and thresholds is also employed. This algorithm was implemented on real time hardware which is described in the next section.

3. REAL TIME HARDWARE IMPLEMENTATION

The speaker verification algorithm discussed above was implemented on special-purpose hardware based around a single Texas Instruments TMS32020 DSP chip [1],[3-5]. Besides the microprocessor, this hardware includes 64K each of program and data memory, a 12-bit A/D converter, and a programmable RS-232 interface. (The system is clocked at 16 MHz.) An IBM-PC serves as the host for the system and functions as a user interface and is also used for template storage. Figure 3 shows a block diagram of this hardware.

The front-end sampling and preprocessing on the TMS32020 are performed using single precision (16-bit) fixed-point arithmetic. The autocorrelation coefficients are computed using a combination of single and double precision (32-bit) fixed-point arithmetic. All other feature and statistical calculations are performed using floating-point arithmetic in which the mantissa and exponent are each represented by a single 16-bit word. The worst-case basic computation times for this latter format are

floating point multiplication	-	5.5 μs,
floating point addition	-	12.8 μs,
floating point division	-	19.2 μs.

Using these critical times as a baseline, it is possible to perform a computational analysis of the verification phase of our speaker authentication algorithm. This is shown below in Table 1. As evidenced by the last line of this table, talker verification operates in approximately 75% of real time. (The training phase of our algorithm does not operate in real time on the current hardware as the updating of 37x37 covariance matrix requires better than three times real time.)

Procedure	Worst-case Times (ms per frame)
A/D Service Routine	3.0
Autocorrelation Computation	4.0
LPC & PARCOR Computation	3.0
Cepstrum & Log-Area Computation	2.5
Feature Selection/Projection	2.8
Updating Statistics	2.3
Liklihood Computation	0.4
Worst-case Prediction	18.0
Actual Performance	14.9

Table 1 - Computational complexity of verification portion of algorithm when operating on custom real time hardware. (*M=4* selected features used for classification.)

The feature selection is of central importance to our algorithm for two reasons. First, the number of computations necessary is reduced from $O(N^2)$ to $O(M^2)$, thereby allowing the algorithm to operate in real time. Secondly, as will be seen shortly, there is no loss in performance by looking only at those parameters which characterize a particular speaker.

4. EXPERIMENTAL RESULTS

The speaker verification algorithm was tested on a population of over ninety speakers. Over seventy seconds of speech for each speaker in a single session, ten seconds of which were used for training. The remainder consisted of fifteen utterances of differing text and were used for verification testing. This provided a total of 90•15 = 1350 type 1 trials and 90•89•15=120,150 type 2 trials. This corresponds to a better than 95% confidence for a 1% confidence interval for each of the verification error rates that follow. (See [1] for a detailed description of the speaker database.)

One important aspect in our algorithm is the improvement in performance as several families of (possibly related) features are combined to form an augmented "raw" feature space. Figure 4 demonstrates this as verification accuracy is plotted as a function of the number of "raw" features. (Several combinations of LPC-based feature families were considered including the LPC, PARCOR, cepstrum, log-area, impulse response of H(z), autcorrelation coefficients, as well as pitch, gain, and zero-crossing rate information.)

A second experiment was to determine the number of selected features which should be used. As Figure 5 indicates, there is no perceivable change in overall system performance after two or three selected features have been used for classification.

Several other similar experiments, too numerous to report here, were conducted . However, once the various parameters of the algorithm were optimized, a final experiment was conducted to evaluate the ultimate performance of the technique. Table 2 gives the results of this experiment. Clearly, these results are promising in light of the modest hardware requirements of this algorithm.

Mode	Type 1 Error Rate	Type 2 Error Rate	Overall Error Rate
Text-Independent	2.3%	1.6%	1.9%
Text-Dependent	1.5%	0.52%	0.94%

Table 2 - Optimized performance of speaker verification algorithm. (M=4 selected features were used.)

Other observations which have been made include the fact that the algorithms appears to be relatively insensitive to the length of time between the training and verification utterances. Also, system performance is unaffected by additive white Gaussian noise down to 15 dB SNR, at which point the speech/silence detector begins to fail.

Finally, equations (2) through (7) indicate that a new feature selection matrix must be computed for each speaker whenever a new talker is enrolled into the system. However, our experience has shown that after ten to twenty speakers have been enrolled, new users can be added without updating all of the $\underline{\underline{A}}_i$'s, while maintaining high performance.

5. CONCLUSION

The paper has described a fast, simple, reliable, yet inexpensive automatic speaker verification system. This system is based around the TI TMS32020 DSP chip and verifies customers in real time.

A new text-independent speaker verification algorithm was implemented on this real time hardware. Several new ideas have been incorporated into this algorithm including the combining of several families of related features, the use of speaker dependent features, and the application of sequential decision logic. All of these help to improve the recognition accuracy of the system.

Although the main emphasis of this paper has been on speaker verification, the ideas presented are also directly applicable to the speaker identification and speaker change detection situations.

ACKNOWLEDGMENTS

The authors wish to acknowledge the many contributions of the RPI speaker verification research group, especially Mr. Daniel Borkowski, Mr. John DellaMorte, and Mr. Michael Elser. We are also grateful to Texas Instruments for donating the TMS32020 development system.

REFERENCES

1. J. B. Attili, "On the Development of a Text-Independent Real-time Speaker Verification System," Ph. D. Thesis, ECSE Dept., Rensselaer Polytechnic Institute, Sept. 1987.
2. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press Inc., New York, 1972.
3. D. G. Borkowski,`"A Real-time Speaker Verification System," M. Eng. Thesis, ECSE Dept., Rensselaer Polytechnic Institute, May 1987.
4. J. D. DellaMorte, "Design of a TMS32020-Based Digital Signal Processor," M. Eng. Thesis, ECSE Dept., Rensselaer Polytechnic Institute, Aug. 1986.
5. M. G. Elser, "Implementation of Speaker Verification Algorithms on the TMS32020" M. Eng. Thesis, ECSE Dept., Rensselaer Polytechnic Institute, May 1987.

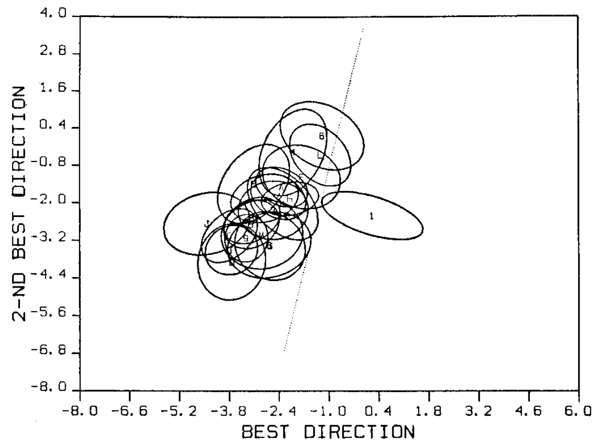

Figure 1 - Scatter of projected features for several speakers projected onto selected subspace for speaker #1.

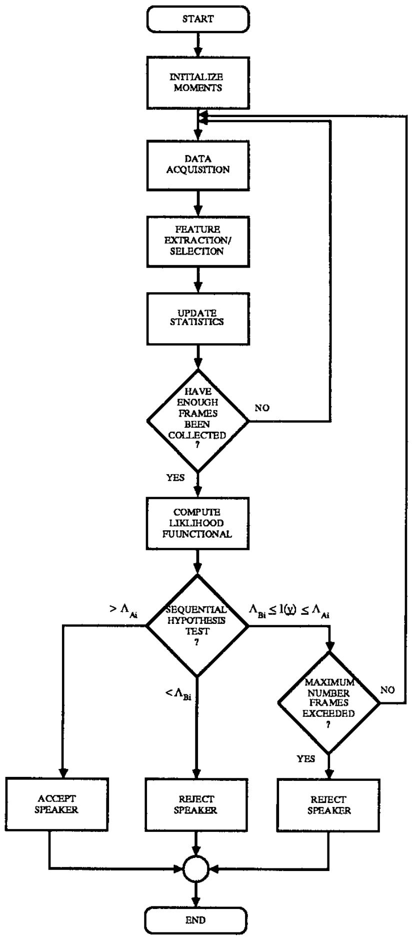
START
INITIALIZE MOMENTS
DATA ACQUISITION
FEATURE EXTRACTION/ SELECTION
UPDATE STATISTICS
HAVE ENOUGH FRAMES BEEN COLLECTED ?
NO
YES
COMPUTE LIKLIHOOD FUUNCTIONAL
SEQUENTIAL HYPOTHESIS TEST ?
$> \Lambda_{Ai}$
$\Lambda_{Bi} \leq l(\underline{y}) \leq \Lambda_{Ai}$
$< \Lambda_{Bi}$
MAXIMUM NUMBER FRAMES EXCEEDED ?
NO
YES
ACCEPT SPEAKER
REJECT SPEAKER
REJECT SPEAKER
END

Figure 2 - Flowchart of verification phase.

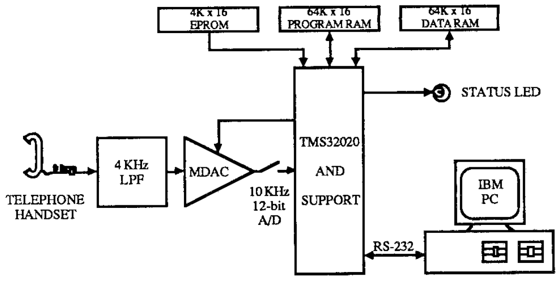

Figure 3 - Block diagram of real time speaker verification TMS32020-based system hardware.

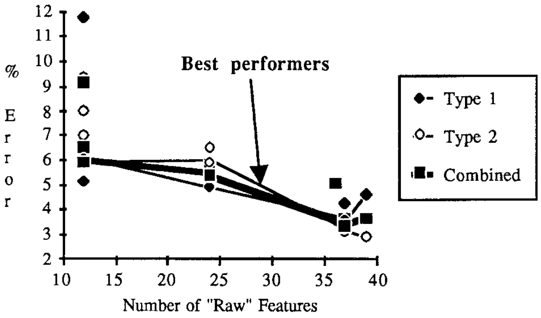

Figure 4 - Improvement of verification accuracy as the number of "raw" features is increased.

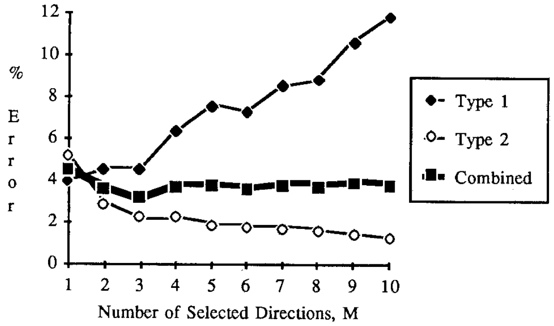

Figure 5 - Verification accuracy as a function of the number of selected features used for classification.