REPORT:

Authors:
John G. Ackenhusen, Syed S. Ali, David Bishop, Louis F. Rosa, *and* ***Reed Thorkildsen*** *are in the Speech Processing department at AT&T Bell Laboratories, Murray Hill, New Jersey.* ***Mr. Ackenhusen*** *is supervisor of the Speech Recognition group. He is responsible for development of efficient algorithms, software, hardware, and silicon chips for real-time speech processing. He received the B.S. (physics), B.S.E. (nuclear engineering), M.S. (physics), M.S.E. (nuclear engineering), and Ph.D. (nuclear engineering) from The University of Michigan. He joined AT&T in 1978.* ***Mr. Ali*** *is a member of the technical staff working on hardware and software for real-time speech processing. He received the B.S. and M.S. in physics from the University of Punjab, Pakistan.*
(continued on page 59)

SINGLE-BOARD GENERAL-PURPOSE SPEECH RECOGNITION SYSTEM

Introduction

This paper describes a single-board implementation of an isolated word recognizer based on the principles of linear predictive coding (LPC) and dynamic time warping (DTW). The recognizer requires only a serial (RS-232) terminal, power supply, and microphone for operation, and may be used to add speech input capability to any serial terminal connected to a host computer. Key elements of the recognizer include a custom integrated circuit for DTW-based pattern matching, a single-chip implementation of real-time LPC feature measurement, and a 16-bit microprocessor for control, communication, and decision functions. As a result of the custom integrated circuit and multiple processor architecture, pattern matching speed is increased by a factor of 50 over an earlier design with no custom integrated circuits and without pipeline processing capabilities, and proceeds on one word while LPC measurement on the next is in progress, increasing speech throughput. Comprehensive control/evaluation software for the recognizer has been developed for the AT&T PC6300 personal computer.

A New System Architecture

A method for speech recognition that combines linear predictive coding (LPC) with dynamic time warping (DTW)[1] has become a standard basis for a wide variety of speech recognition systems and has been systematically optimized over the past 10 years. Careful tests have used experienced and inexperienced talkers speaking over dialed-up telephone lines to examine the performance of most aspects of the recognition algorithm. The tests included speaker-trained isolated word recognition,[1] speaker-independent isolated word recognition,[2] connected word recognition,[3] methods of endpoint detection,[4,5] techniques of dynamic time warping,[6] and procedures for training.[7] Other tests have embedded the recognizer in systems that used vocabulary partitioning, directory searches, and syntactic analysis to perform such voice-activated tasks as repertory dialing of telephone numbers,[8] retrieving telephone directory information,[9] and making airline reservations.[10] In all simulations, the technique was shown to attain performance sufficient for practical use over telephone lines for a wide variety of talkers.

Accompanying the continual refinement of the LPC/DTW algorithm has been an evolution of real-time hardware implementations of the algorithm. A special-purpose computer for performing the LPC/DTW isolated word recognition algorithm in real time included a two-board customized processor similar to present-day single-chip digital signal processors that was used to perform the numerically demanding computations of LPC and DTW.[11] A formal study was conducted to examine parameters of the LPC/DTW algorithm that would be impacted by modification for real-time implementation.[12]

To provide a smaller, faster, and less costly implementation of the algorithm, a new system architecture was devised that consisted of a dedicated single-chip processor for LPC

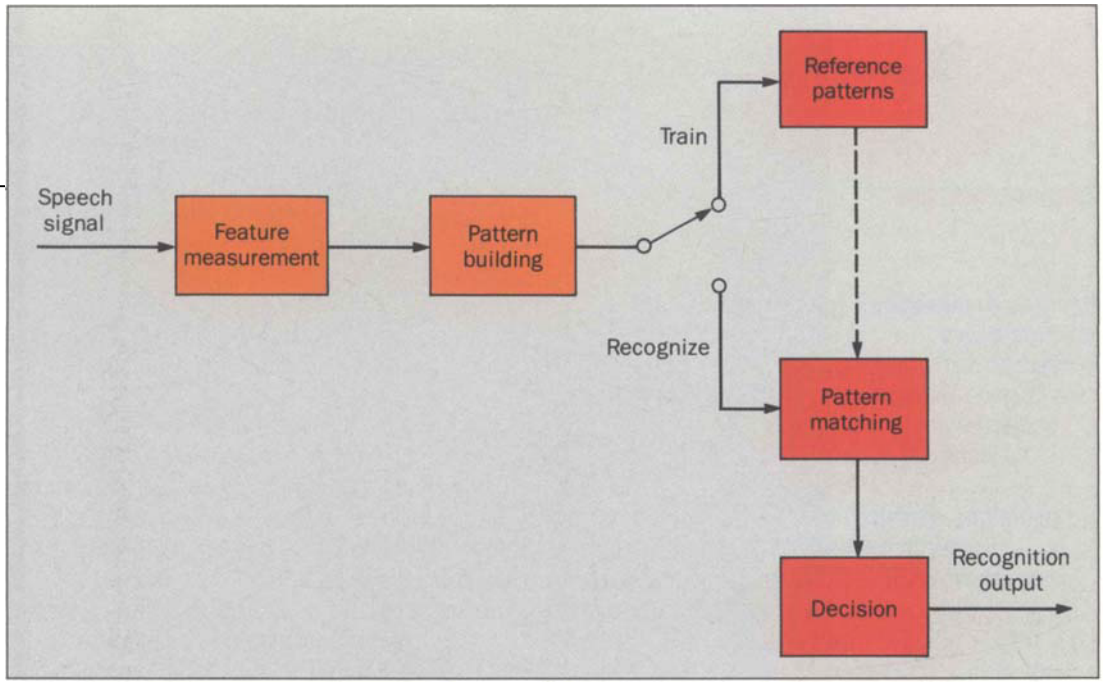

Figure 1. Speech recognition system.

computation, a second single-chip processor for DTW computation, and a third general-purpose microprocessor for all remaining control and communications operations. This architecture increased speech throughput by tailoring a processor for each stage of the algorithm and allowing each processor to run independently and in parallel with the others. The result, described in this paper, was a single-board recognizer (SBR) capable of performing word pattern matches at the rate of 1000 patterns per second and completing the entire process of recognizing a word from a vocabulary of 150 words in less than 300 ms. This is a significant improvement over the earlier five-board system that could perform word pattern matches at a 22 pattern per second rate and recognize one word in a 40-word vocabulary in 900 ms.

In the following section, we describe the recognition algorithm implementation. A subsequent section describes the control firmware and operating system that reside on the board, as well as the evaluation and control program package that runs on the AT&T PC6300 personal computer for application development, diagnosis, and demonstration.

Recognition Algorithm Implementation

A simplified block diagram (Figure 1) of this speech recognition system includes a feature measurement block, a pattern-building section, a pattern-matching section, and an output decision block.

In the feature measurement section, the input speech signal is broken up into 45-ms segments known as frames, and a feature vector is computed for each frame on a characterization of the short-time spectrum by linear predictive coding. During training of the recognizer, a reference vocabulary of word patterns is built by concatenating the sequence of feature vectors and storing them in memory. In the recognition mode, unknown word patterns are constructed, compared to each reference pattern in memory, and scored for similarity. The decision block searches the similarity scores and reports recognition results.

The speech recognition task is performed on the SBR in three processors which operate in parallel (Figure 2). The first is an AT&T Technologies DSP-20 programmable dig-

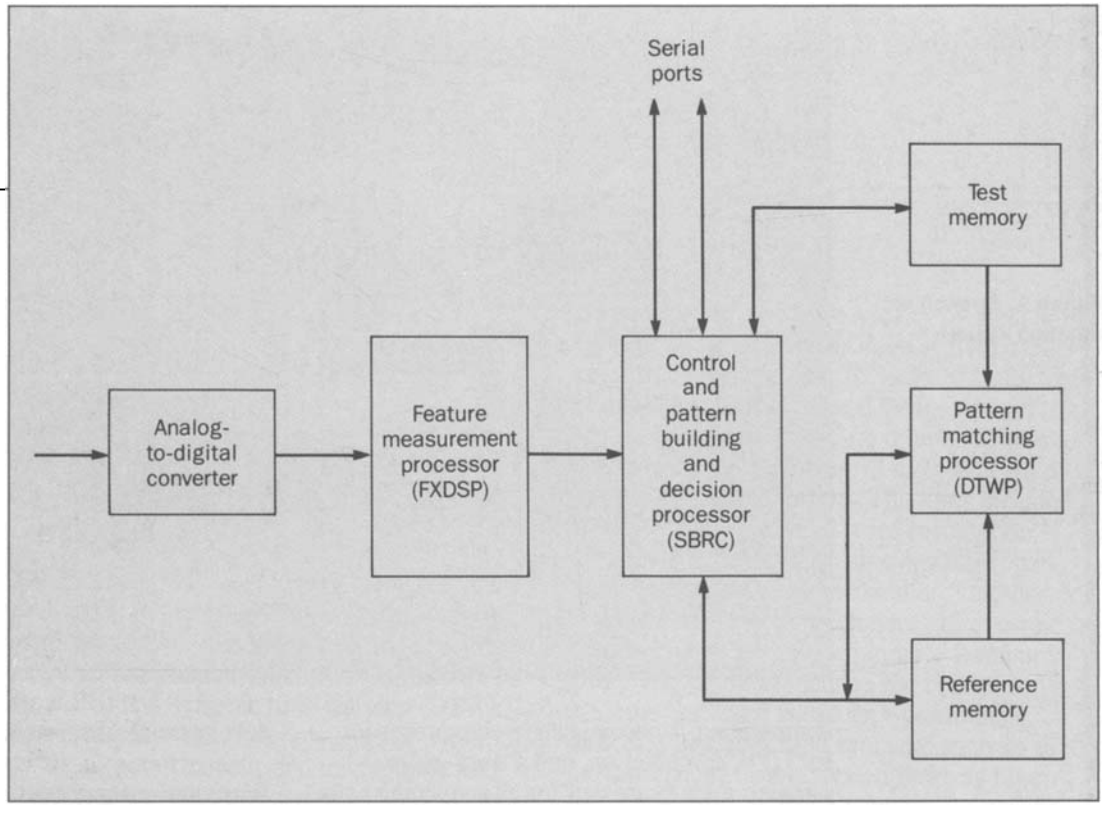

Figure 2. Architecture of single-board recognizer.

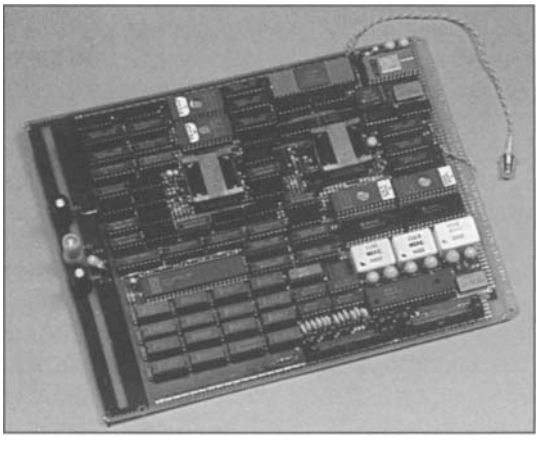

Figure 3. Single-board recognizer hardware.

ital signal processor performing feature measurement. The input to this processor, called the feature-extraction digital signal processor (FXDSP), is μ-law PCM-encoded samples of the speech signal. LPC feature vectors from the FXDSP are then stored in a circular buffer.

The second processor, called the single-board recognizer controller (SBRC), is the processing element in the pattern-building/control block and is implemented with a 16-bit microprocessor. The SBRC takes feature vectors out of the circular buffer, determines the beginning and end points of words, and builds test or reference word patterns. In the recognition mode, it also loads the test and reference cache memories of the pattern-matching section with the appropriate word patterns and controls the operation of the dynamic time warp processor (DTWP).

The pattern-matching block contains the third processor, the DTWP. It is an application-specific integrated circuit which computes dissimilarity scores (distances) of the unknown input pattern (test) from previously stored patterns (references). The distances are passed back to the control block for decision processing. Finally, the SBRC presents the recognition results to a host computer or ter-

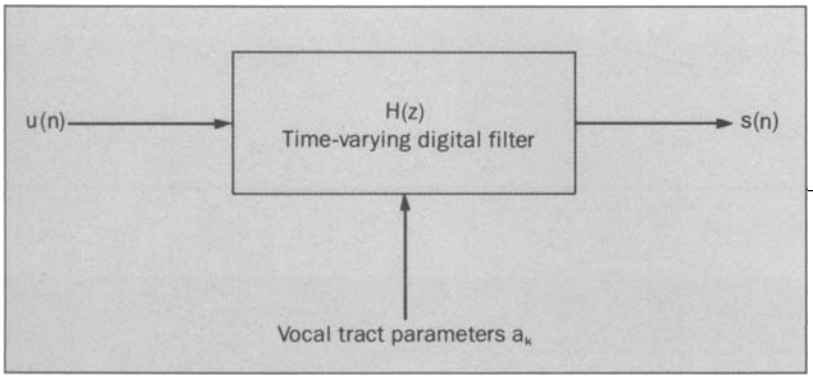

Figure 4. Speech model including excitation function and transfer function.

minal over serial ports (Figure 2). The SBR accommodates its three processors and other components on an 8 by 10 inch board (Figure 3).

Feature Measurement. A spectral model for each frame of speech is computed by means of linear predictive coding. This model of the speech waveform is based on the idea that the speech sample can be predicted by a linear combination of past speech samples. The model is represented by a time-varying digital filter of order p with an all-pole transfer function of the form

$$H(z) = \frac{1}{1 - \sum_{k=1}^{k=p} a_k z^{-k}}$$

The LPC model as implemented in the FXDSP uses $p = 8$. This filter is excited with the excitation function $u(n)$, and the speech samples $s(n)$ are related to the excitation $u(n)$, in Figure 4, by

$$s(n) = \sum_{k=1}^{k=p} a_k s(n-k) + u(n)$$

A linear predictor with prediction coefficients α_k is defined as a system whose output is

$$\tilde{s}(n) = \sum_{k=1}^{k=p} \alpha_k s(n-k)$$

The prediction error, $e(n)$, is defined as the difference between the true signal $s(n)$ and the predicted signal $\tilde{s}(n)$:

$$e(n) = s(n) - \tilde{s}(n)$$

The feature vector components which are the result of LPC analysis on a frame of speech are based on the coefficients α_k, which minimize the total prediction error over the analysis frame.

The FXDSP is a real-time implementation of the LPC feature-measurement technique using the AT&T DSP20 programmable single-chip digital signal processor.[13]

Telephone bandwidth (150 to 3200 Hz) speech is digitized by a μ-law coder-decoder (codec) with filters at a 6.667-kHz sampling rate. This digitized signal is passed to the FXDSP processor, which performs an eighth-order LPC analysis and logarithm of energy computation on the signal in real time. An analysis frame size of 300 samples (45 ms) is used with a new frame beginning every 100 samples (15 ms). Thus, each speech sample falls within three consecutive analysis frames. Every 15 ms, the FXDSP processor completes the LPC analysis of one frame and puts out a feature vector consisting of the log energy, nine amplitude-normalized autocorrelation coefficients, and nine LPC-based test coefficients for that frame (Figure 5).

Formatting (SBRC). The FXDSP runs continuously, outputting feature vectors which represent the signal at its input whether that signal is speech, silence, or noise. An important process in the speech recognition task is the formatting of a continuous stream of feature vectors into word patterns. The first step in the formatting process is real-time endpoint detection based on the energy temporal contour of the signal. Several parameters obtained from a measurement of the background noise energy level in the recognition environment are

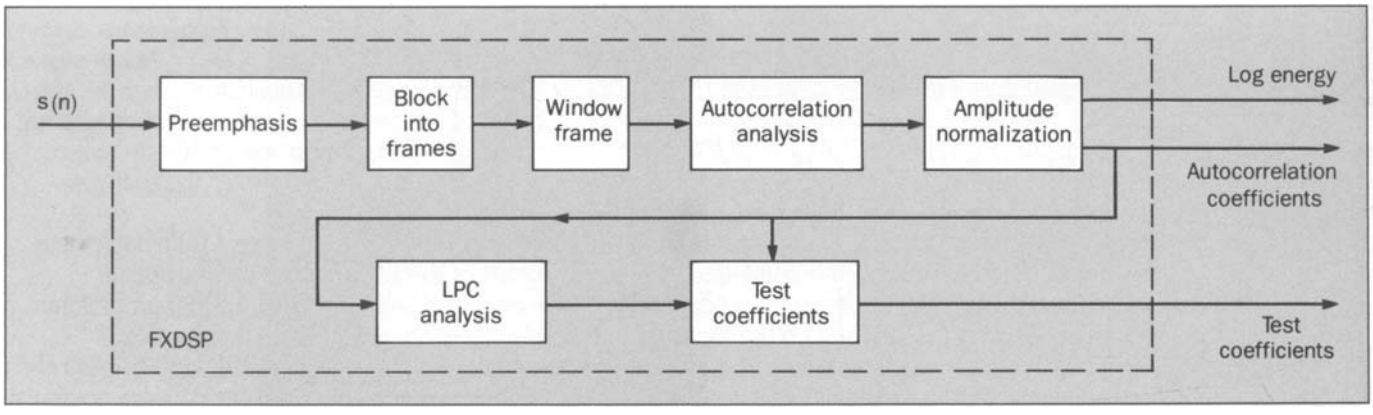

Figure 5. Signal processing to extract linear predictive coding features for recognition.

required. The background noise level is determined either in a static mode by sampling for a fixed period of time while no speech is present[4] or in a dynamic fashion by developed by J. F. Lynch, Jr., of Bell Laboratories involving continuous updating with each new log energy.

In the static mode, a histogram of the samples of background noise energy is computed. A three-point median smoother is applied to this histogram. The background level is the mode of the smoothed histogram.

When dynamic computation of the background level is selected, each new log energy is processed by a low-pass filter. The rise-time constant of the filter is chosen to be longer than the duration of a typical word so that the background level is not appreciably changed during speech, but will ultimately reflect any real change in the background noise energy level. The fall-time constant of the filter is short so that between words, the background level tracks the lowest input log energy. In this way the background noise energy level is updated with each new log energy.

Once the endpoint parameters are computed, the frames representing the approximate beginning and end of a word are marked. The word is stored in a circular buffer for further processing. The circular buffer, as its name implies, allows buffering of multiple word patterns while the rest of the recognition steps are being performed on a word. As word patterns are taken out of the buffer, the next formatting step is refinement of the word endpoints.[5,6] After endpoints have been determined, the word is length-normalized to 40 frames[4] to create the word template that will be used as the test pattern during pattern matching.

Pattern Matching (DTWP). The pattern matching block is based on the dynamic time warp processor (DTWP) integrated circuit.[14] A pattern for an unknown utterance is compared to stored reference patterns by the time-alignment method of dynamic time warping. Dynamic time warping is the process by which the effects of nonlinear variations in speaking rate are minimized. For example, in Figure 6a, even though the lengths of the energy temporal contours of the two words are the same, their peaks do not align because of nonlinear variations in speaking rate. By means of a time-

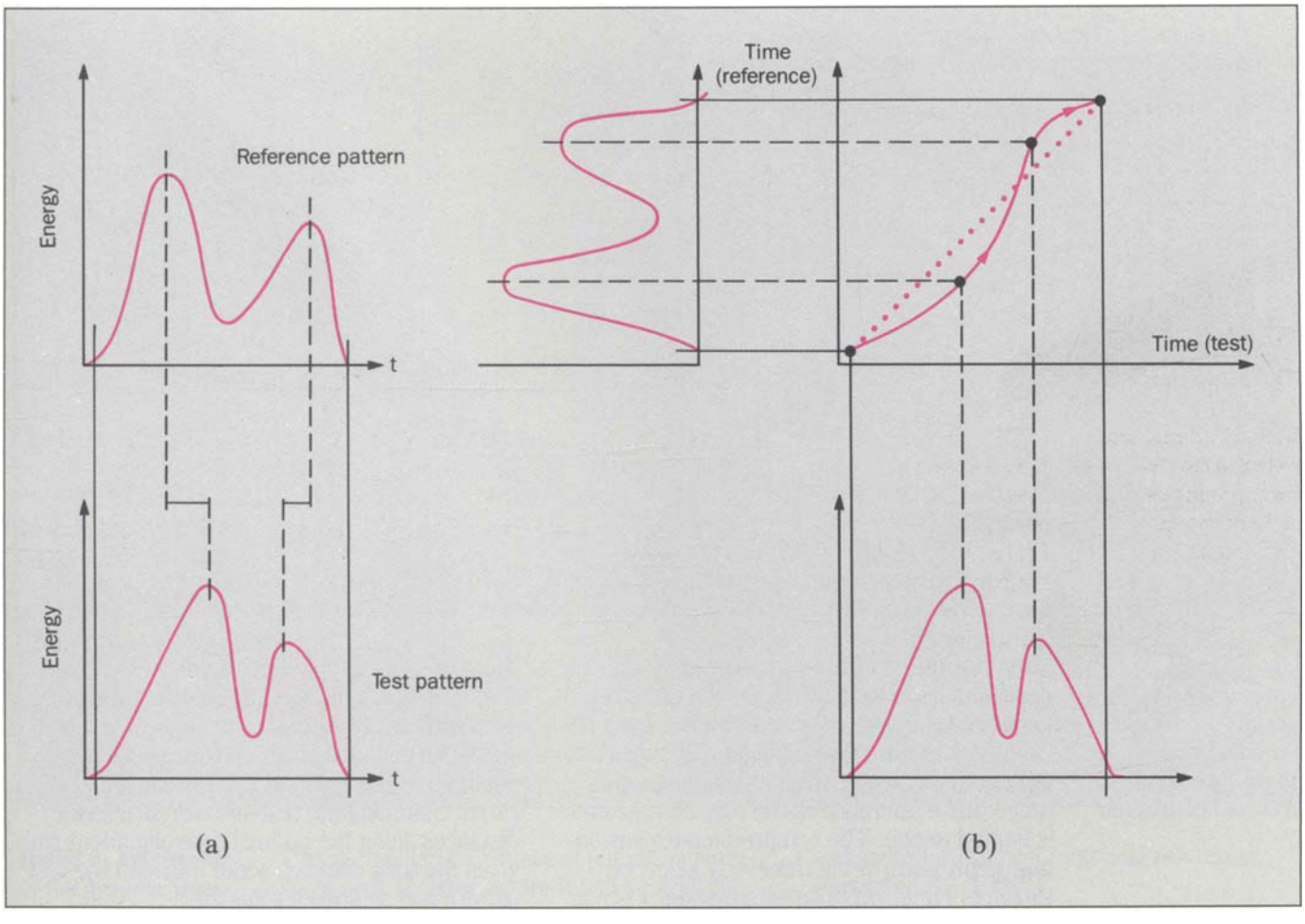

Figure 6. Dynamic time warping. (a) Unaligned peaks. (b) Time-aligned path that deviates from linear alignment.

alignment path that deviates from linear alignment, the peaks may be aligned and the pattern similarity may be computed along this optimum path (Figure 6b). The pattern-matching block performs all the necessary arithmetic and decision-making operations for selecting a word from a given vocabulary, comparing a test pattern with up to 256 reference patterns using the technique of dynamic time warping to optimally align the time axis of each reference pattern to the test.

To eliminate unreasonable departure from linear mapping of the test and reference frames, several constraints are placed upon the alignment. The endpoints of the test and reference are required to match. The reference

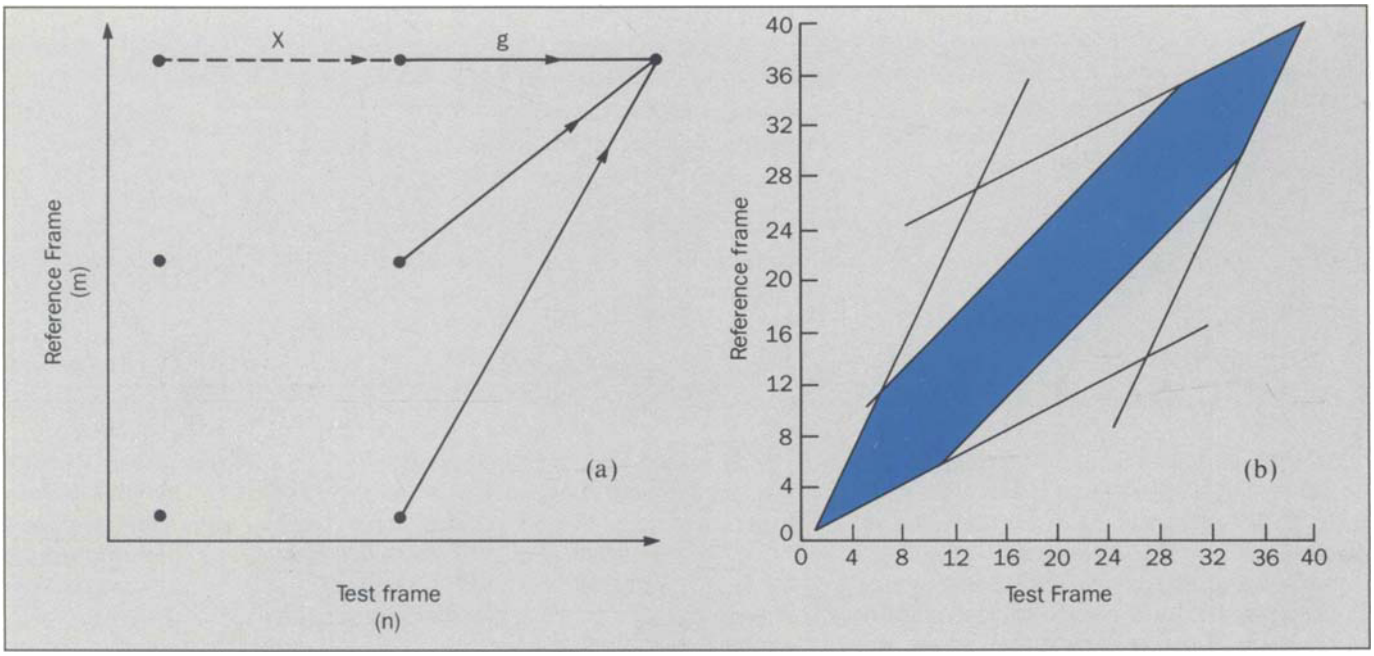

Figure 7. Effects of constraints on dynamic time warping. (a) Compression-expansion constraint. (b) Global constraint.

utterance may be time-compressed by skipping one frame for each test frame, time-mapped linearly to the test, or time-stretched by duplicating a reference frame (alignment path slope of 2, 1, or 1/2, respectively). The number of times that a reference frame may be repeated is limited to one. The compression-expansion constraint leads to the three-way selection shown in Figure 7a. The path marked x is not allowed. By adding the final global constraint that the deviation of the optimal path from the diagonal be limited to ± 5, the one-to-one time mapping of test frames to reference frames is restricted to the parallelogram of Figure 7b.

Comparison of a test frame to a reference frame requires a measure of closeness. The Itakura log likelihood ratio[1] distance function yields a measurement that is indicative of the spectral energy difference between the two frames of speech. By appropriately computing a set of test and reference coefficients, the log likelihood distance between frames can be obtained as the result of a dot product and logarithm. Summing the test-to-reference frame distances along the optimal time alignment path gives the total distance score between the test pattern and reference template.

The entire DTW algorithm is reflected in the functional block diagram of the DTWP, Figure 8. At each possible point along the time warp path, a local distance d_{ij} is computed from the ith test and jth reference coefficients. The accumulated distance to this point is $D(i,j)$ such that

$$D(i,j) = d_{ij} + \min [D(i-1, j), D(i-1, j-1), D(i-1, j-2)]$$

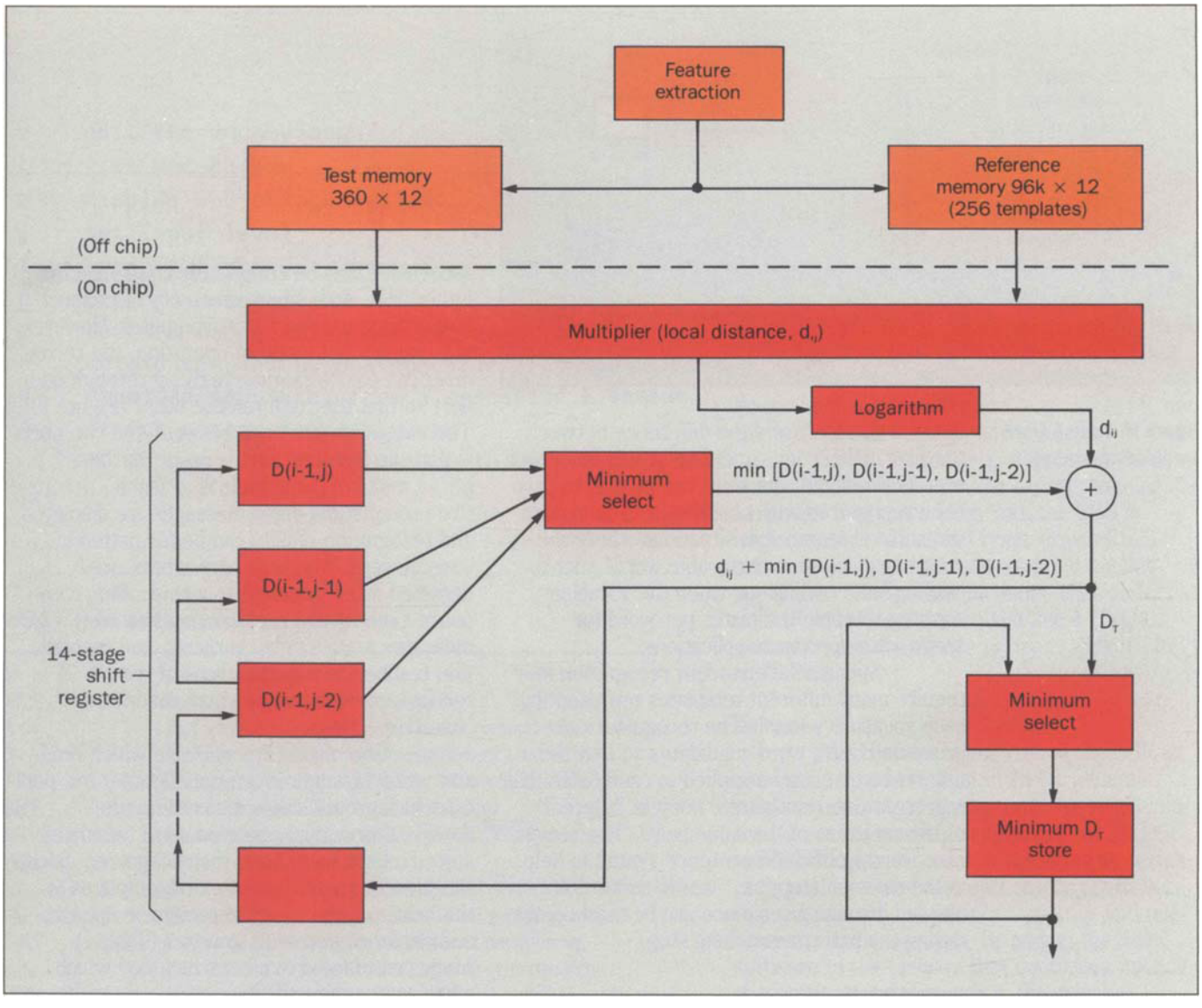

Figure 8. Architecture of the dynamic time warping processor.

This accumulated distance is put into a multistage shift register where it will be available for the distance calculations between the following test frame and the reference frames. The accumulated distance at the end of the time warp path D_T will be the score for the optimal alignment path between the test pattern and the reference template. The DTWP can sequentially compute scores between the test pattern and up to 256 reference templates, outputting the scores and/or retaining the index and score for the best matching reference (minimum D_T store).

Decision. After the pattern-matching block has computed distance scores between each of the reference templates and the test pattern, a confidence test is applied to determine whether the best scoring reference candidate is reliable. The SBR uses a distance threshold and a separation requirement to make this decision.

If the top candidate distance score is

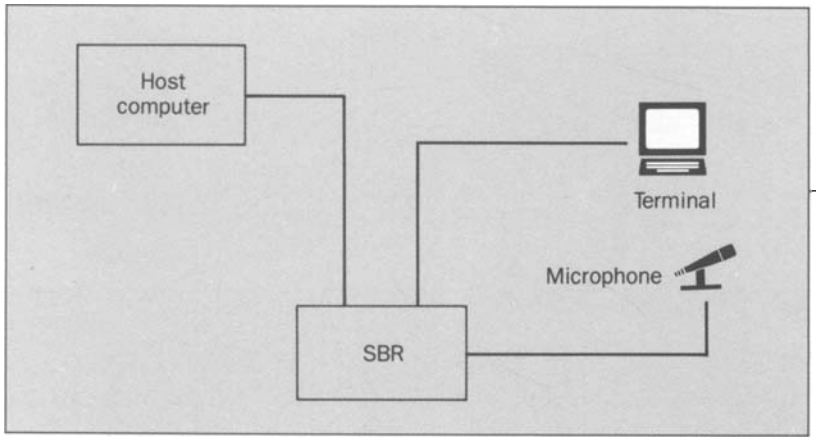

Figure 9. Transparent mode of operation.

below a threshold and the difference between the first and second candidate scores is above a second threshold, the word can be said to have been recognized with confidence. The second threshold is required in situations where the vocabulary contains confusable words such as "sailing" and "saline" or when the vocabulary contains multiple templates per word for speaker-independent applications.

Speaker-independent recognition may require many different templates representing each vocabulary word. The recognizer can group and mask word candidates so that the difference test can be applied to candidates that represent different words, not just different representations of the same word. The recognizer can also include sentence syntax to help in the decision step; i.e., words that are not allowed in a word sequence can be masked out during the pattern-matching step.

Control Interface

SBRMON Monitor. The SBRMON monitor program allows external control of the recognizer and data flow over two RS232 ports. The three main blocks of this section take care of hardware and parameter initialization after a reset of the system, command line interpretation or downloading of system information, and output of system status, results, or system parameters.

One of the two RS232 ports is designed to interface to a host computer while the other port interfaces to a terminal. The recognizer may be completely controlled by either port, so it is not necessary to connect both a terminal and a host computer. However, in a transparent mode of operation, the recognizer can pass all signals arriving through one port so that they exit out the other (Figure 9). The essential difference between the two ports is that the terminal port is more "verbose"; i.e., a menu of commands is available, descriptive prompts and error messages are displayed, and recognition results can be formatted in several ways. The host port syntax was designed as an interface to a controlling computer. Commands, prompts, and error messages are a single character, and recognition results are simply indices of the recognized words rather than descriptive character strings.

Commands are available which read and write various recognition parameters, perform background noise measurements, download previously created word templates stored on the host, train the recognizer, upload the present vocabulary of word templates to the host, tell the board to recognize speech, and perform system diagnostics (Table I). Many commands have been included which allow customizing the recognition algorithm to the application environment. For example, it is possible to vary word endpoint parameters and read out the resulting boundary points in speech as the words are processed.

Host Control/Evaluation Program (SBRCOM). SBRCOM is a software package designed to run on the AT&T PC 6300 and compatibles. It provides a menu-driven interface to the SBR. Figure 10 presents a summary of initial commands that appear on the computer screen as options to the user. SBRCOM allows the user to easily perform complex operations with the

```
v  - edit the vocabulary
l  - list vocabulary
o  - display and change options
g  - set input level
t  - train the present vocabulary
p  - perform recognition
e  - perform recognition and evaluate results
u  - upload templates to a specified filename
d  - download templates from a specified filename
h  - direct access to recognizer
^q - control-q - quit and return to DOS
```

Figure 10. SBRCOM menu.

Table I. Host and Terminal Port Commands

M	Display SBR command menu
RES	Reset hardware, initialize parameters
A	Perform automatic background noise measurements (dynamic)
B	Perform background noise measurement (static)
TR	Training mode; up to 142 words, 40 words at a time
RET	Retrain a word in the current vocabulary
UT	Upload templates from SBR to host (PC6300 or other)
WR	Write recognition parameters
D	Download-append templates from host system (PC6300 or other)
DG	Download group table for template set grouping
DMT	Download mask table for template set partitioning
P	Recognition; no template set grouping or partitioning
PG	Recognition; template set grouping with template partitioning

SBR such as modifying recognition parameters, entering vocabulary, training, scoring recognition, and uploading and downloading templates. Using SBRCOM, one can perform tests of recognition performance by playing prerecorded utterances through the board and storing the results of recognition attempts in the host computer. SBRCOM also supplies important diagnostic information such as plotting of the log energy contour and word boundaries found by the recognizer as a result of endpoint detection. Figure 11 shows an SBRCOM plot of the logarithm of signal energy vs. time, the endpoints of the word (vertical lines), and the energy thresholds used in endpoint detection (horizontal lines). Each major increment, e.g., 40 to 50, on the vertical axis corresponds to 12 dB.

SBRCOM consists of a series of software modules written in the C programming language. Communication with the SBR is implemented with an interrupt-driven RS232 software package that was modified for this application. The communications uses serial port COM1, interrupt request line 4 (IRQ4), and automatically adjusts to baud rates of 110 to 19,200.

Summary

Much of the board area of the SBR is consumed by a fairly typical 16-bit general-purpose microcomputer system. As an alternative to a stand-alone board design, the FXDSP and DTWP processors may be added as peripherals to the central microprocessor controller that may already reside in a computer and telecommunications system. By adding the SBRC software to the tasks of that controller, speech recognition capability may in this way be included in a tightly integrated manner with minimum incremental cost.

We have described a subsystem for isolated word recognition known as the single-board recognizer. The recognizer uses the mature LPC and DTW algorithms, optimized for real-time implementation. Speech throughput is increased by partitioning of the speech recognition task into feature analysis, pattern matching, and control. Each subtask is performed on a separate processor which runs independently and in parallel with the other

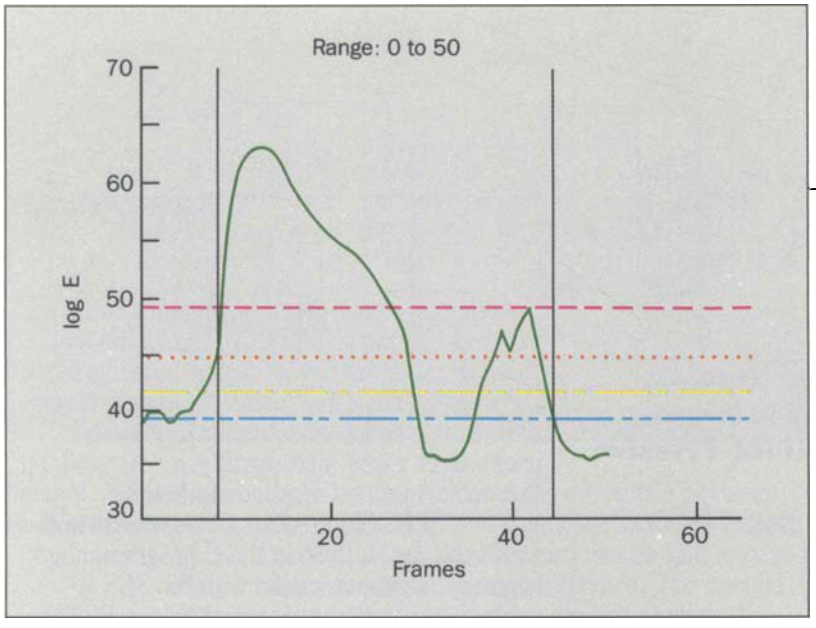

Figure 11. Plot of the logarithm of energy.

processors on the board.

Of particular significance to the speech processing capabilities of the SBR are the FXDSP, a single-chip implementation of real-time LPC feature measurement and the DTWP, a custom integrated circuit for high-speed DTW-based pattern matching. A high-performance 16-bit microprocessor coordinates the operation of the FXDSP and DTWP and handles I/O processes for the SBR.

The SBR and its control program, SBRCOM, have been in internal exploratory use within AT&T for the past few years. In addition, the SBR performs speech recognition in both the "Lost for Words" exhibit at Walt Disney World's EPCOT Center and in the "Mouse in a Maze" exhibit at the InfoQuest demonstration center at AT&T headquarters in New York City.

Acknowledgment

The work of H. F. Carbonneau, H. N. Carlson, J. G. Josenhans, J. Kumar, and M. Lalumia, culminating in the final implementation of the pattern-matching integrated circuit and its transfer to manufacture, was essential to the realization of the single-board recognizer described here. It is a pleasure to acknowledge the valuable work of these colleagues.

References

1. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-23, February 1975, pp. 67-72.
2. L. R. Rabiner et al., "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-27, August 1979, pp. 336-349.
3. C. S. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-29, April 1981, pp. 284-97.
4. L. F. Lamel et al., "An Improved Endpoint Detector for Isolated Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-29, August 1981, pp. 777-785.
5. J. G. Wilpon, L. R. Rabiner, and T. Martin, "An Improved Word Detection Algorithm for Telephone Quality Speech Incorporating Both Syntactic and Semantic Constraints," *AT&T Bell Laboratories Technical Journal*, Vol. 63, No. 3, March 1984, pp. 479-498.
6. C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-28, December 1980, pp. 622-635.
7. L. R. Rabiner and J. G. Wilpon, "A Simplified, Robust Training Procedure for Speaker-Trained, Isolated Word Recognition," *Journal Acoustical Society of America*, Vol. 68, No. 5, November 1980, pp. 1271-6.
8. L. R. Rabiner, J. G. Wilpon, and A. E. Rosenberg, "A Voice-Controlled Repertory Dialer System," *Bell System Technical Journal*, Vol. 59, No. 7, September 1980, pp. 1153-1163.
9. B. Aldefeld et al., "Automated Directory Listing Retrieval System Based on Isolated Word Recognition," *Proc. IEEE*, Vol. 68, No. 11, November 1980, pp. 1364-79.
10. S. E. Levinson and K. L. Shipley, "A Conversational-Mode Airline Information and Reservation System Using Speech Input and Output," *Bell Sys-*

tem Technical Journal, Vol. 59, No. 1, January 1980, pp. 119-137.
11. J. G. Ackenhusen and L. R. Rabiner, "Microprocessor Implementation of an LPC-Based Isolated Word Recognizer," *Proceedings*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 1981, pp. 746-749.
12. L. R. Rabiner, J. G. Wilpon, and J. G. Ackenhusen, "On the Effects of Varying Analysis Parameters on an LPC-Based Isolated Word Recognizer," *Bell System Technical Journal*, Vol. 60, No. 6, July-August 1981, pp. 893-911.
13. J. G. Ackenhusen and Y. H. Oh, "Single-Chip Implementation of Feature Measurement for LPC-Based Speech Recognition," *AT&T Technical Journal*, Vol. 64, No. 8, October 1985, pp. 1787-1805.
14. M. K. Brown et al., "The DTWP: An LPC-Based Dynamic Time-Warping Processor for Isolated Word Recognition," *Bell Laboratories Technical Journal*, Vol. 63, No. 3, March 1984, pp. 441-457.

Biographies (continued)

Mr. Bishop *is a consultant working on hardware and software interfaces for speech recognition products. He joined AT&T in 1984. He received the A.S.E.E. degree from Thames Valley Technical College, Connecticut, and is working toward the B.S.E.E. degree at Rutgers University.* ***Mr. Rosa*** *is a senior technical associate working on firmware and models for the single-board speech recognizer. He joined AT&T in 1980. He received the A.O.S. degree in electronics circuits and systems from the Technical Careers Institute, New York.* ***Mr. Thorkildsen*** *is a member of the technical staff responsible for design and development of speech recognition subsystem hardware and software. He joined AT&T in 1981. He received the Ph.D. in physics from the University of Virginia.*

(Manuscript received August 26, 1986)