AT&T BELL LABORATORIES TECHNICAL JOURNAL
Vol. 63, No. 3, March 1984
Printed in U.S.A.

An Improved Word-Detection Algorithm for Telephone-Quality Speech Incorporating Both Syntactic and Semantic Constraints

By J. G. WILPON,* L. R. RABINER,* and T. MARTIN*

(Manuscript received August 26, 1983)

Accurate location of the endpoints of spoken words and phrases is important for reliable and robust speech recognition. The endpoint detection problem is fairly straightforward for high-level speech signals in low-level stationary noise environments (e.g., signal-to-noise ratios greater than 30-dB rms). However, this problem becomes considerably more difficult when either the speech signals are too low in level (relative to the background noise), or when the background noise becomes highly nonstationary. Such conditions are often encountered in the switched telephone network when the limitation on using local dialed-up lines is removed. In such cases the background noise is often highly variable in both level and spectral content because of transmission line characteristics, transients and tones from the line and/or from signal generators, etc. Conventional speech endpoint detectors have been shown to perform very poorly (on the order of 50-percent word detection) under these conditions. In this paper we present an improved word-detection algorithm, which can incorporate both vocabulary (syntactic) and task (semantic) information, leading to word-detection accuracies close to 100 percent for isolated digit detection over a wide range of telephone transmission conditions.

I. INTRODUCTION

In an automatic speech recognition system, it is assumed that during a recording interval (which may be continuous) a user will speak a command, which he wants the recognizer to interpret and respond to

* AT&T Bell Laboratories.

accordingly. The first task of a recognition system is to separate the input speech from the various types of nonspeech events that also occur during the recording. This task is referred to as endpoint detection.

Accurate detection of the spoken word has been shown to be crucial for reliable word recognition.[1,2] Most research in the study of designing endpoint detectors has used speech databases, where the speech has been collected over clean transmission mediums [using close-talking, noise-canceling microphones, or telephone speech over local Private Branch Exchanges (PBXs)]. The signal-to-noise ratio (s/n) under these conditions is high (between 35 and 50-dB peak s/n). Also, the noise generated in such a system is usually stationary. This research has led to quite reliable endpoint detectors.

Endpoint detection becomes much more difficult when the transmission system is corrupted by the many noises one finds on a standard, dialed-up telephone line. Some of these problems include popping sounds, crackling noises, carrier frequency tones, background speech, and other nonstationary noises. The need for an accurate speech endpoint detector that works as well in these environments as in clean environments is a goal that has not been met. In an earlier study,[1] when telephone customers, speaking over randomly dialed telephone lines with various types of transmission distortion, were asked to speak their telephone number as a sequence of isolated digits, existing endpoint algorithms often failed.

To evaluate a new endpoint scheme, we must define the requirements on endpoint accuracy. An indirect measure of these requirements can be obtained directly from the recognizer as follows. Given a test set of many spoken words, use them as input to a word recognition system consisting of an endpoint detector and a recognizer. If, when substituting a new endpoint detection algorithm for an earlier one, we obtain higher word recognition accuracy, then we will say that the new endpoint detection algorithm is better than the earlier one.

One way of explicitly defining the requirements on endpoint accuracy is to perform the following experiment. Take a speech database of isolated words and manually detect the beginning and end of each word. Next, vary the beginnings and ends of each word over some specified range (e.g., ± 150 ms) and perform isolated word recognition. By examining sensitivity of the recognition scores to variability in endpoints, an explicit relationship can be found. Such an experiment was performed and will be described in this paper.

The purpose of this paper is to describe a new approach for determining the endpoints of spoken words, which incorporates both vocabulary (syntactic) and task (semantic) information, leading to word-detection accuracies close to 100 percent for isolated digit detection over a wide range of telephone conditions. We call the new approach a top-down design. Simply put, we look for strong (vowel-like) peaks in the energy contour of a speech utterance and process the speech around the peaks to find potential beginning and ending points. Several rules involving duration, and onset and decay times are then used to refine the endpoint estimates.

This new algorithm is compared to an earlier endpoint algorithm by Lamel et al.,[2] which tries to find word endpoints based on the energy of the speech rising some fixed level above the background noise energy. We call this type of approach a bottom-up approach. In addition we will briefly identify several other algorithms that were investigated, none of which performed as well as the top-down approach.

The format of this paper will be as follows. In Section II we review the bottom-up approach to endpoint detection. We describe our new top-down word detector in Section III. In Section IV we describe the database used in all our tests, present results on the tests to explicitly measure requirements for word-detection accuracy, and give recognition results comparing the bottom-up approach to the new top-down method.

II. REVIEW OF BOTTOM-UP ENDPOINT DETECTOR

Figure 1 gives a block diagram of the bottom-up endpoint algorithm of Lamel et al.[2] First the input speech is bandpass filtered and sampled. Since we are working with a telephone bandwidth signal, we bandpass the speech signal from 100 to 3200 Hz and sample it at a 6.67-kHz rate. The digitized speech is then preemphasized using a simple first-order digital filter with a z transform:

$$H(z) = 1 - az^{-1}, \qquad (1)$$

where $a = 0.95$. The digitized speech is then blocked into frames of N samples, with a shift between frames of L samples. Experimentation has found that N should be set to 300 samples and L should be set to 100 samples. This corresponds in time to 45-ms frames with a 15-ms shift between frames. Each frame of speech is then weighted by a Hamming window of the form

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right), \qquad 0 \leqslant n \leqslant N-1 \qquad (2)$$

(where N is previously defined). Windowing reduces the truncation effects of the framing procedure. We denote the ℓth frame of windowed speech as $s_\ell(\text{n})$ defined for $0 \leqslant n \leqslant N-1$.

After this initial digital processing, the energy of the signal is

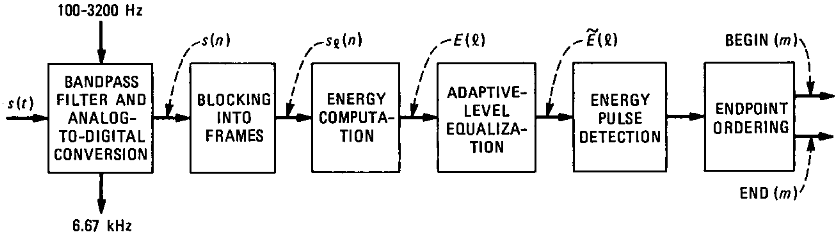

Fig. 1—Block diagram of bottom-up endpoint algorithm of Lamel et al.

computed. This computation can be made simply by summing the squares of the signal values during a frame of speech. However, since we are using a Linear Prediction Coding (LPC) recognizer,[3-9] which requires that a pth-order autocorrelation analysis (in our case $p = 8$) be performed on the entire recording interval, the energy is extracted as a by-product of the analysis. That is,

$$E(\ell) = 10 \log_{10} R_\ell(0), \qquad \ell = 1,2, \cdots, NF, \tag{3}$$

where NF is the total number of frames in the recording interval, $R_\ell(0)$ is the zeroth-order correlation coefficient,

$$R_\ell(0) = \sum_{n=0}^{N-1} [s_\ell(n)]^2, \tag{4}$$

and $E(\ell)$ is on a decibel scale.

The next step in the processing (called adaptive-level equalization in Fig. 1) is a normalization of the energy contour to compensate for the mean background noise level. First, $E_{\min}$ is computed as

$$E_{\min} = \min_{1 \leq \ell \leq NF} (\mathrm{E}(\ell)). \tag{5}$$

$\hat{E}(\ell)$ is then formed as the difference between $E(\ell)$ and $E_{\min}$,

$$\hat{E}(\ell) = E(\ell) - E_{\min}, \qquad \ell = 1,2, \cdots, NF. \tag{6}$$

Next, the background level estimate is refined even further by computing a histogram of the signal energies. The histogram is restricted to the lowest NP dB (typically $NP = 15$) of $\hat{E}$. We then apply a three-point median smoother to this histogram. Finally, we create the modified energy contour $\tilde{E}(\ell)$, $\ell = 1,2, \cdots, NF$,

$$\tilde{E}(\ell) = \hat{E}(\ell) - \text{Mode}, \tag{7}$$

where Mode is the mode of the smooth histogram generated above.

The remaining blocks of the bottom-up endpoint detector are the energy pulse detector and an endpoint ordering procedure. The energy pulse detector scans the modified energy contour and selects all potential energy pulses within the recording interval. Pulse-combining rules are used to eliminate short pulses, and combine close pulses. Several parameters, along with their current settings, need to be defined in order to explain how these blocks operate. These parameters include:

1. $K1$, $K2$, and $K3$ are energy thresholds used in determining the word boundaries (3, 10, 5 dB).

2. IT1 and IT2 are frame counter thresholds for determining the presence or absence of any breath noises at the boundary points of a detected utterance (5,5 frames).

3. IT3 is the minimum length for a detected pulse (5 frames).

4. NFMIN is the minimum length in frames for an utterance (10 frames).

Figure 2 shows a state representation of the operation of the energy pulse detector. The normalized energy ($\tilde{E}$) of the recording is scanned from left to right ($\ell = 1$ to $\ell = NF$). If $\tilde{E}(\ell)$ rises first above K_1, then above K_2 (without falling below K_1), a beginning pulse marker is assigned to frame ℓ. Similarly, when the energy dips below K_3 an ending marker is assigned. The beginning IT1 frames and ending IT2 frames are checked for breath-type noises (i.e., low energy content throughout the IT1 or IT2 frames), and eliminated if necessary. All pulses must have a minimum length (IT3). Pulses are then combined based on their proximity to other pulses. All final pulses are checked for duration and maximum energy content, and pulses that do not pass are eliminated. The final output of the endpoint detector is a set of ordered pairs of beginning and ending points of segments within the recording interval. It is assumed that each segment corresponds to a spoken word within the recording interval. The Lamel et al. bottom-up endpoint detector can be, and has been, implemented as a real-time endpoint detector.

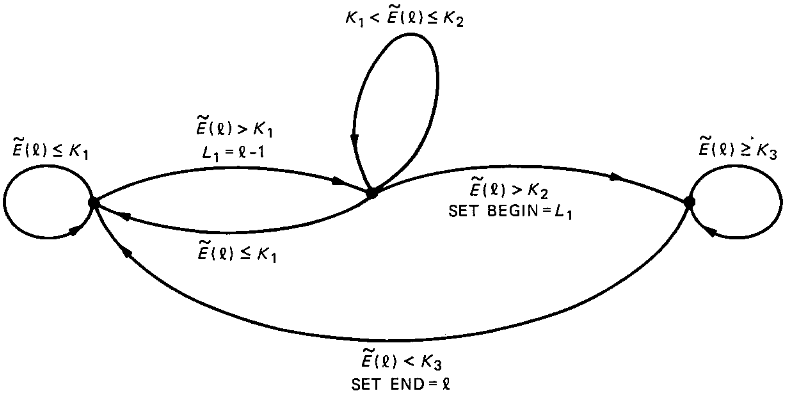

Fig. 2—State representation of energy pulse detector from Lamel et al. endpoint detector.

III. DESCRIPTION OF THE TOP-DOWN ENDPOINT DETECTOR

As discussed previously, the bottom-up endpoint detector works very well in stationary noise backgrounds with reasonably high signal-to-noise ratios. However, in highly variable noise background conditions it tends to fail at a very high rate. Hence, we now describe a top-down approach capable of finding words in highly nonstationary backgrounds.

The design of the top-down endpoint detector is similar to that of the bottom-up approach in that it computes a normalized energy array, finds pulses in the recording interval, and then combines them to get the final endpoint decisions. The differences lie in the energy pulse detection and endpoint processing procedures.

To understand the differences, we need to define some additional parameters along with their current settings, namely:

1. MXWD is the number of utterances within a recording interval (7 words).

2. IGAP is the number of frames from which a pulse slope is computed (3 frames).

3. ISLOPE is the pulse slope threshold (7 dB).

4. NSEP, NSEP2 are pulse separation counters (2,7 frames).

Figure 3 gives a flow diagram of the energy pulse detection proce-

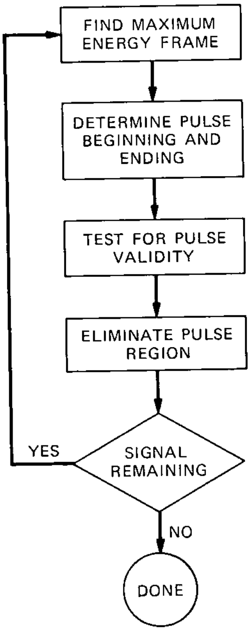

Fig. 3—Block diagram describing energy pulse detection procedure from top-down endpoint algorithm.

dure. The philosophy is to find the high energy frames in a local region and then try to define the energy pulse boundaries using the lower energy frames. In particular, the algorithm scans the entire recording interval (i.e., $\hat{E}(\ell)$, $\ell = 1,2, \cdots, NF$) until it finds the frame with the highest energy. The algorithm then analyzes the energy values of the surrounding frames. It looks at frames prior to the maximum energy frame until it finds a frame with energy less than the threshold $K1$, and it looks at frames beyond the maximum energy frame until it finds a second frame with energy less than the threshold $K3$. At this point the pulse detector has found a set of possible beginning and ending frames for an utterance—i.e., an energy pulse. Its next task is to try to eliminate any breath noises at the estimated boundaries of the energy pulse. This is performed by testing the first IT1 frames and last IT2 frames of the energy pulse for consistently low energy content. Next, the detected pulse (corresponding to the utterance) is checked to guarantee that its duration is greater than a minimum-length threshold and that its amplitude is above a minimum level. Pulses are eliminated if they do not pass these tests. This procedure is iterated throughout the recording interval. All previously detected pulses are eliminated from consideration in each new iteration. When this process is complete, a set of NPULSE pulses are found within the recording interval. Figure 4a shows a typical energy plot of a string of isolated digits indicating where pulses were detected. In this example, six energy pulses were detected; however, there are only four spoken digits in the recording interval.

The energy pulses are next sent to a pulse combiner algorithm, which attempts to combine two or more adjacent pulses to form longer pulses. This process works as follows. First, all pulses are sorted in order of decreasing peak energy. We then start with the pulse with the highest peak energy and try to add pulses to it based on the following rules.

For a prior pulse to be added to the beginning of the current energy pulse, first the Downward Slope (DS) (defined over the last IGAP frames of the pulse) of the pulse must be above a threshold. Such a sharp downward slope tends to occur during stop-gap-type pulses within a word. Second, the prior pulse must lie within NFW frames of the current pulse (where NFW is determined by DS). If these conditions occur, the prior pulse will be combined with the current pulse to give a single combined pulse. In a similar manner, a pulse can be added to the end of the current energy pulse. In addition to the slope constraint, there are other restrictions for combining pulses. The duration of the combined pulse must be below a maximum-length threshold. (Clearly, if the combining pulse duration is too long, it signifies that two distinct words were spoken close to each other, and

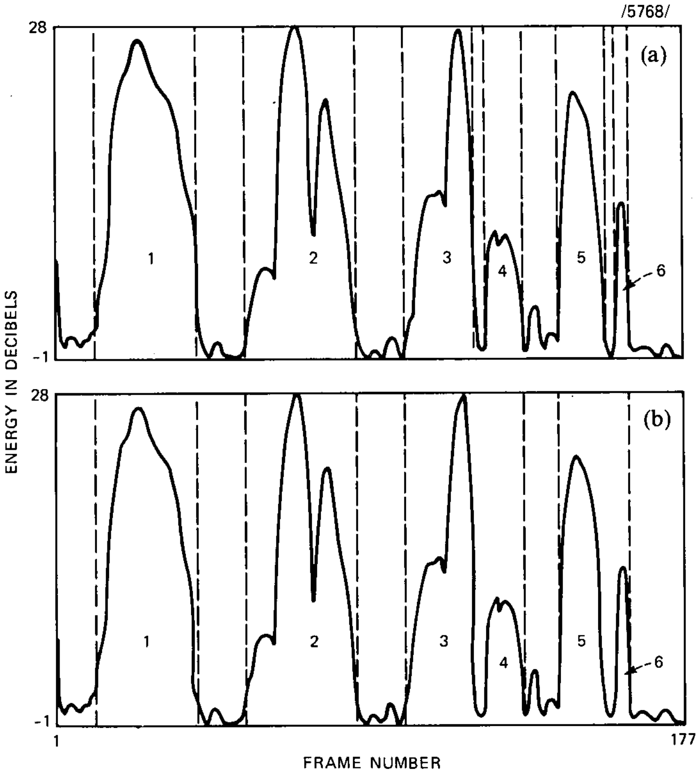

Fig. 4—Log energy contour from spoken string of isolated digits. Dashed lines indicate (a) where pulses were detected and (b) result of applying pulse combiner rules to pulses found in (a).

hence, should not be combined.) This restriction is not applied when the algorithm is detecting connected words as well as isolated word sequences. A second restriction is that the upward slope value (defined similarly to the downward slope value) between two combining pulses must also be above a threshold. This situation typically signifies a stop gap within a word. Figure 4b shows the result of applying the pulse combiner rules to the spoken sequence of Fig. 4a. Pulses 3 and 4 have been combined (this is the digit *six*), as have pulses 5 and 6 (this is the digit *eight*).

3.1 Syntactic constraints of digits

In the final decision block, the first task is to eliminate all endpoint utterances that are too short (i.e., less than the threshold NFMIN). The algorithm could terminate here, with the final output being a list of beginning and ending frame pointers for each detected utterance. However, we have incorporated several decision rules based on the knowledge that we are detecting digit strings. For this special vocabulary, only two words (the digits *six* and *eight*) can possibly contain a stop gap. Thus, all other words in the vocabulary can be represented by a single energy pulse with no other pulses attached. Also, for the digits *six* and *eight*, the maximum energy pulse is always the first pulse when a secondary pulse is added. Given the rules for combining pulses, this implies that no pulse should be added to the beginning of a maximum energy pulse. Next, both the digits *six* and *eight* have at most only one stop gap present, implying that at most one pulse can be added to the end of a maximum energy pulse. By adding these additional rules to the endpoint detector, we can increase overall accuracy for this specialized vocabulary.

3.2 Semantic constraints from the digit recognition task

We further assume that the input speech is a sequence of MXWD isolated digits (e.g., MXWD equals seven for a telephone number). Thus, for this specialized case, we know the number of utterances within the recording interval. This information can also be incorporated into the algorithm. One way to implement this idea is to sort the final endpoint detector output in order of maximum peak energy level and to retain the top MXWD utterances. If the output of the pulse combiner indicates that fewer than MXWD words were found, we assume that some of the uttered words were spoken as connected sequences rather than as isolated words. This is because the pulse detector has its parameters set to find any spoken utterance with a peak energy of greater than 10 dB (note the average peak energy for utterances recorded previously is between 30 and 50 dB[2,6]).

IV. EVALUATION OF THE TOP-DOWN APPROACH TO ENDPOINT DETECTION

To evaluate the top-down endpoint detector, a series of experiments were performed using telephone recordings from a subset of the data described in Ref. 1. This database consisted of 11,035 digits spoken by 3153 people in highly variable telephone transmission conditions. For evaluation purposes we used a subset of 820 digits spoken by 218 talkers. This particular subset of data was used because its statistics were similar to those of the entire 11,035-digit database. Also, the experiments we planned to perform were so computationally extensive we wanted to choose a small subset of the database.

For recognition purposes we used the 30-template-per-digit reference set used in Ref. 1. These templates were extracted from a subset of 3700 digit tokens using the Unsupervised Without Averaging (UWA) clustering algorithm.[3]

The first experiment concerned direct measurement of recognition accuracy as a function of error (as measured with respect to hand-chosen endpoints) in endpoint location. The second set of experiments compared the bottom-up and top-down approaches on the 820-word test vocabulary.

4.1 Recognition as a function of endpoint location error

Based on the energy contour of the recording interval and on careful listening to the speech, we manually determined the endpoints for each of the 218 strings (to the nearest 15-ms interval). Figure 5 shows some typical speech utterances along with their manually determined endpoints. These examples show some of the typical problems associated with endpoint detection of this database. Figure 5a, which shows the energy contour for the string /391/, exhibits a nonstationary noise floor, where a person was talking in the background. Figure 5b, which shows the contour for the digit string /8292/, exhibits transients that were introduced by the transmission system (i.e., P_1, P_2, P_3, P_4 are the transients, while A_1, A_2, A_3, A_4 are the actual spoken digits). Note that the peak s/n is 31 dB, but the s/n's of most of the individual digits are

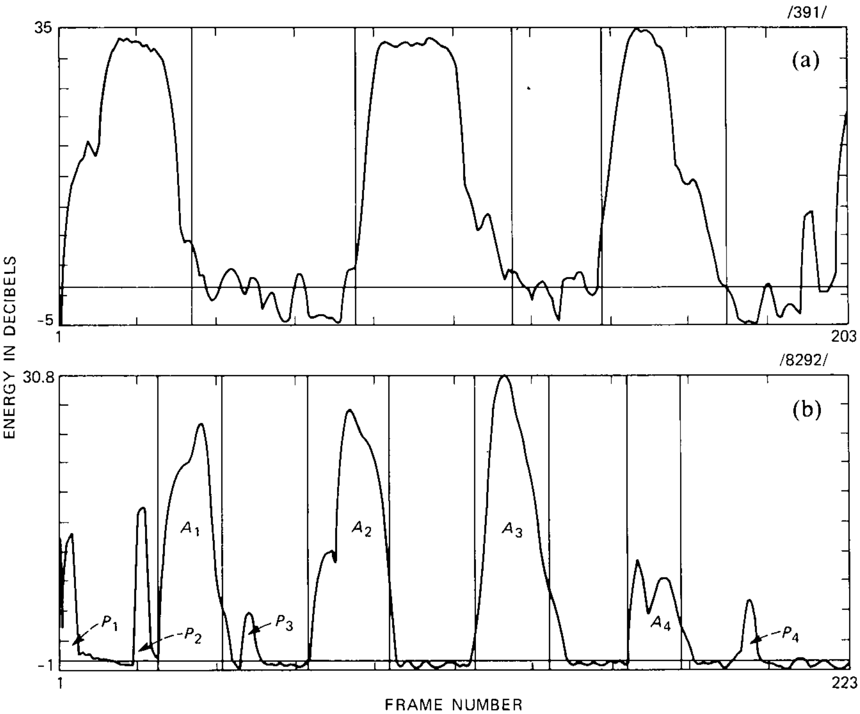

Fig. 5—Log-energy contour for the spoken strings (a) /391/ and (b) /8292/. Solid lines indicate manual placement of word endpoints.

lower (e.g., the second digit, *two*, has peak s/n of about 12 dB). After the endpoints were manually determined, recognition was performed on the isolated digit database. The recognizer used was the LPC-based recognizer,[3-8] which has been used and studied extensively at AT&T Bell Laboratories. A simple K-nearest neighbor decision rule was used in all tests. The overall recognition accuracy obtained was 93.0 percent using the manual endpoints.

After recognition was performed, the manually detected beginning point and ending point of each word were automatically varied in 15-ms (single-frame) steps from 150 ms before the manually determined endpoint to 150 ms after the endpoint. Recognition was performed at each interval with the results tabulated in the form of a contour plot. Figure 6 shows a contour plot of overall recognition accuracy as a function of the change in the endpoint position in ms. Each ring represents a 1-percent change in recognition accuracy. The contour plot was obtained by averaging over all digits in the test database. Figure 6 shows, as anticipated, that the best recognition score, 93.0 percent, was obtained when the exact manually determined endpoints were used.

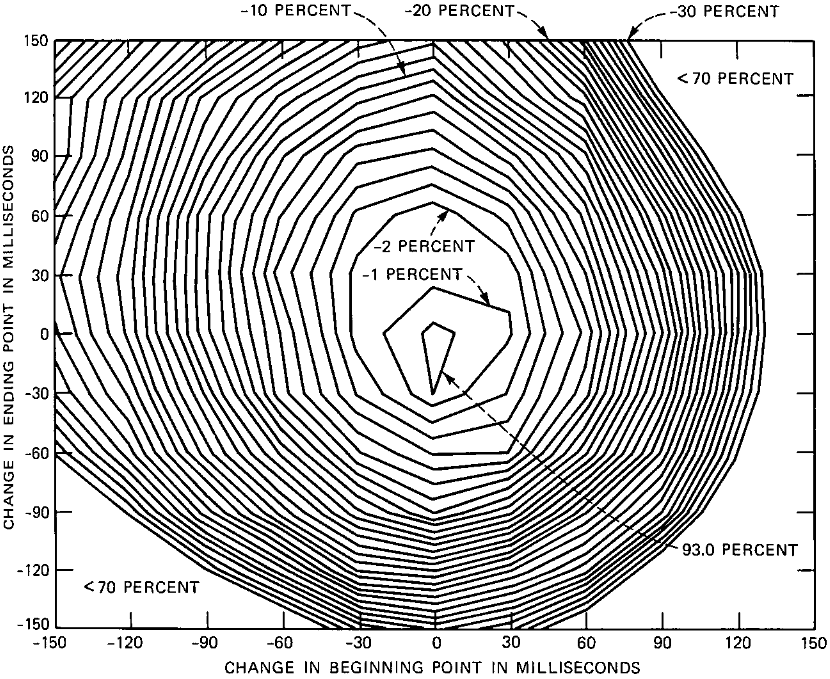

Fig. 6—Contour plot showing results of recognition experiment where manually placed endpoints were varied by ±150 ms. Results are averaged over all digits. Each ring represents a 1-percent change in recognition accuracy.

The contour plot of Fig. 6 also implies that if the endpoints were varied only slightly from the hand-placed endpoints, the recognition accuracy would drop. For example, a 3-percent reduction in accuracy occurred if both the endpoints were in error by ±60 ms. We see that the rings of the contour plots are fairly concentric, implying a uniform decrease in recognition accuracy as the endpoints are placed further away from manually chosen ones.

If we look at contour plots of the individual digits, we see that their rings are definitely not concentric, and that the best recognition accuracy from most of the digits was not obtained using the manually determined endpoints. Table I gives the best recognition scores on a per-digit basis, along with the changes that were made to the endpoints in order to obtain those results. Figures 7 through 9 show contour plots for some of the digits. We can make several observations from these curves. Figure 7 shows the contour plot for the word *zero*. The best accuracy for this word (averaged over all occurrences of the word) was obtained if the manually determined beginning points were moved in (i.e., closer to the ending point) by 30 ms. It can be seen that the digit *zero* is more sensitive to variations in the ending point than the beginning point. For the digit *one* (see Fig. 8), we see the best results (96.1 percent) were obtained if the ending point was moved out by 90 ms (six frames). This is quite a large amount, and may be justified by the fact that the nasal sound at the end of the word *one* is of such low energy that, using the energy contour and human listening, accurate placement of the ending point cannot be made. This plot also shows that the beginning point of the digit *one* is much more sensitive than the ending point. If the beginning points are varied by −60 ms (from the optimal point), the recognition accuracy drops by 28 percent, but if the ending point is varied by −60 ms, the recognition accuracy drops only 1.5 percent. Figure 9 shows the contour plot for the word *six*. We

Table I—Recognition results from test run with modified endpoints

Digit	Percent Correct	Change in Beginning Point (ms)	Change in Ending Point (ms)
0	90.4	30	0
1	96.1	0	90
2	94.9	0	−60
3	95.6	0	0
4	93.8	0	−30
5	96.7	0	0
6	95.7	0	−90
7	97.2	−30	30
8	97.1	0	−30
9	88.9	0	30
Total over all digits	93.0	0	0

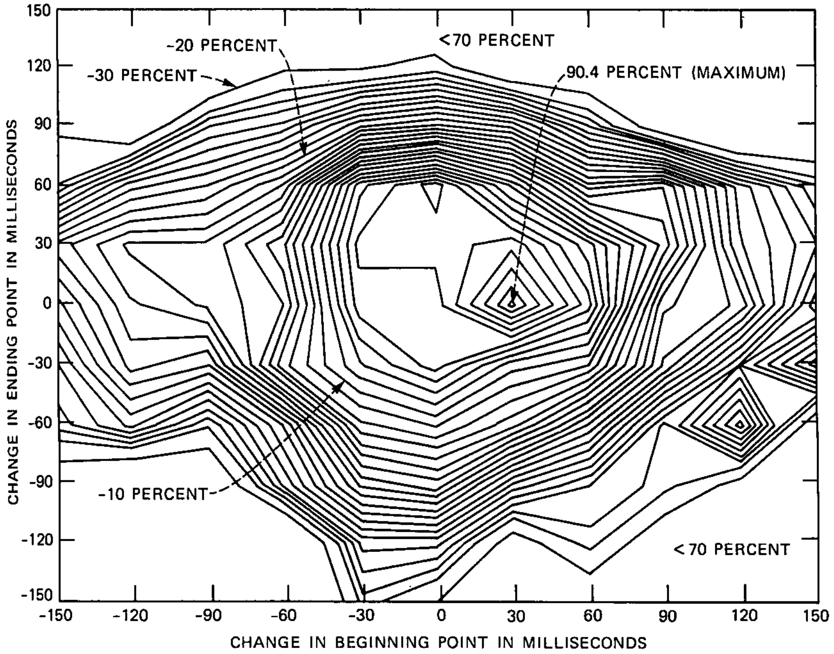

Fig. 7—Contour plot showing results of recognition experiment where manually placed endpoints were varied by ±150 ms. Results are only for the digit *zero*. Each ring represents a 1-percent change in recognition accuracy.

see that the rings are highly nonuniform, with several local maxima present throughout the plot. The best recognition accuracy for *six* was obtained when the ending points were cut back by 90 ms (95.7 percent). However, a cutback in the ending points of only 30 ms coupled with a 30-ms increase in the beginning points also yielded the same results (95.7 percent). Similar observations can be made for the rest of the digits.

The main point to emphasize is that extremely accurate determination of the speech endpoints must be made, in order to obtain the highest system accuracy using our LPC-based recognition system.

4.2 Accuracy of automatically determined endpoints

Endpoints were automatically determined for the 820-digit database using both the bottom-up and top-down approaches. Figure 10 shows a histogram of the error in frame location of the top-down endpoints compared to the manually determined endpoints for the 820 digits. Figure 10a shows results for the beginning frame; Fig. 10b shows results for the ending frame. The automatically determined endpoints agree with the manual endpoints within ±1 frame 68.2 percent of the

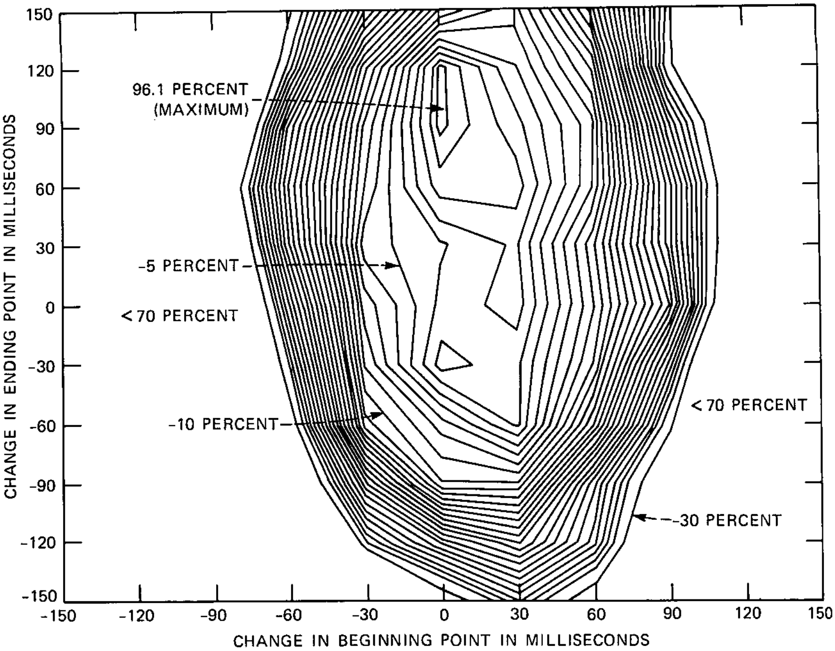

Fig. 8—Contour plot showing results of recognition experiment where manually placed endpoints were varied by ±150 ms. Results are only for the digit *one*. Each ring represents a 1-percent change in recognition accuracy.

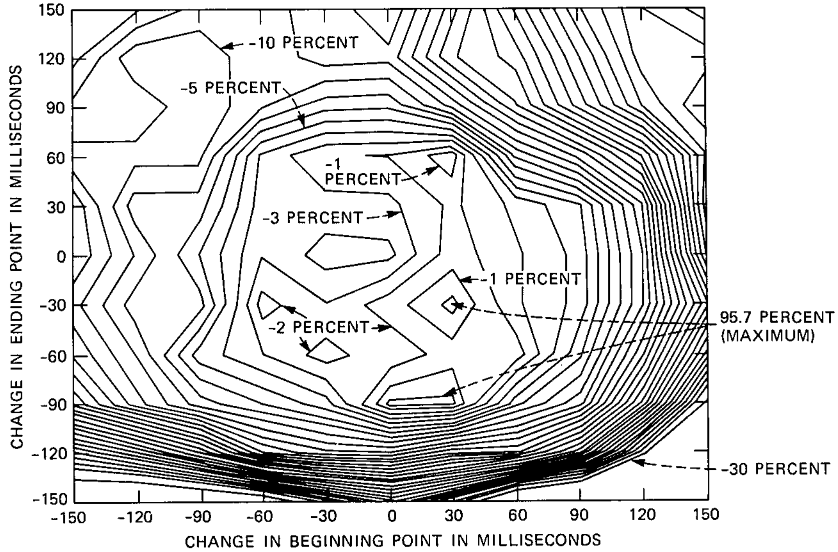

Fig. 9—Contour plot showing results of recognition experiment where manually placed endpoints were varied by ±150 ms. Results are only for the digit *six*. Each ring represents a 1-percent change in recognition accuracy.

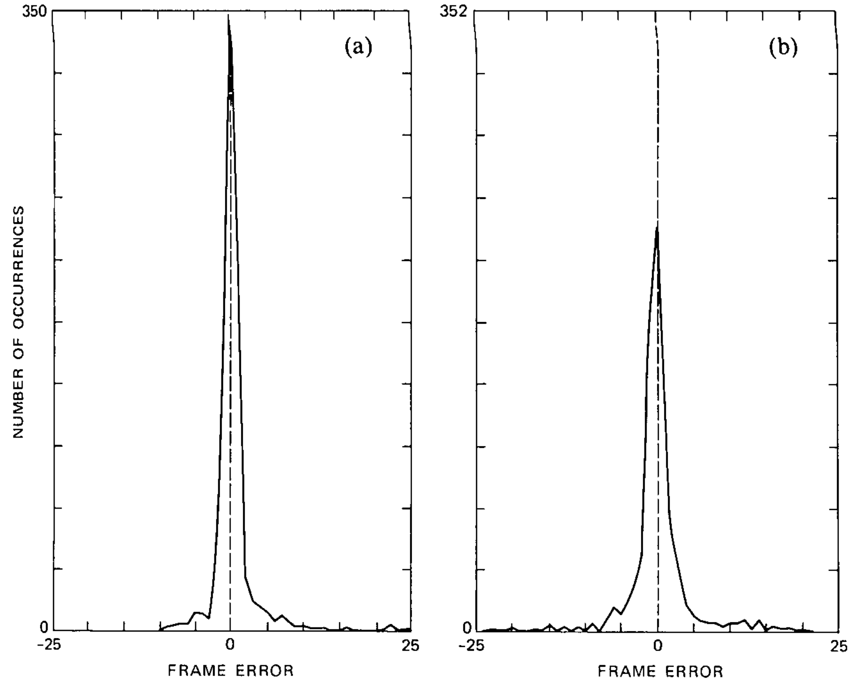

Fig. 10—Histogram of error in frame location of top-down endpoints compared to manually determined endpoints for (a) beginning frame, and (b) ending frame.

time, within ±2 frames 78.5 percent, and within ±5 frames 90.0 percent for the combined beginning and ending points.

Figure 11 shows some examples of how the new automatic endpoint detector worked on several representative strings of digits. Shown are log-energy contours with dashed lines indicating where the algorithm determined the digit endpoint locations to have been. The string in Fig. 11a, /2226242/, is an example of speech spoken in the presence of highly variable background noise. The peak s/n for this example is 21.7 dB; however, the s/n's for most of the digits in the string are well below that figure. Under laboratory conditions (speech over a local PBX) the peak s/n is usually between 35 and 50 dB. The string in Fig. 11b, /6854566/, is an example of how the endpoint detector is sometimes able to split connected words (the 68 and 4566 are connected). The string in Fig. 11c, /4736354/, shows a fully connected string of digits. The first three digits /473/ were determined to be one utterance, and the next four digits, /6354/, were split into separate utterances. Finally, the string in Fig. 11d, /2294761/, shows that the new endpoint detector can work very well even on very bad background conditions. Note the extremely variable background noise level (peak s/n of only 17 dB).

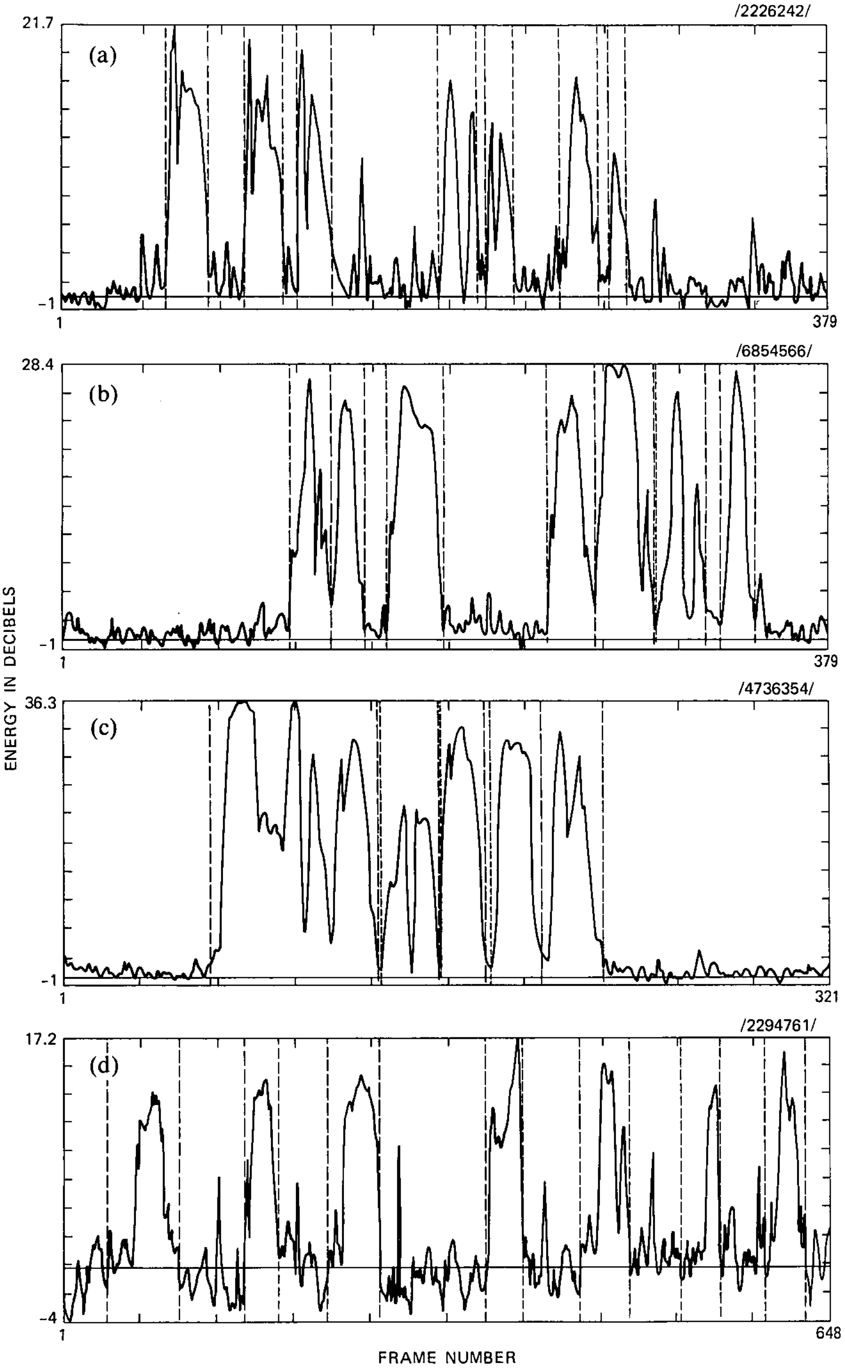

Fig. 11—Log-energy contours from several representative digit strings showing how top-down endpoint detector performed. Dashed lines indicate where algorithm placed beginning and ending word markers. (a) String /2226242/ shows speech spoken over highly variable background noise. (b) String /6854566/ shows how endpoint detector can split connected words. (c) String /4736354/ shows a fully connected string of digits. (d) String /2294761/ shows speech carried over very noisy telephone lines.

4.3 Recognition results using the bottom-up approach

Recognition was run on the 820 digits database using the bottom-up endpoint detector. For this algorithm, only 68.5 percent of the 820 digits were detected. Of the detected words, 85.2 percent were correctly recognized.

4.4 Recognition results using the top-down approach

The top-down endpoint detector was substituted for the bottom-up approach and the entire recognition process was repeated. The new endpoint detector found 800 of the 820 digits (97.6 percent). In looking further we found that 10 double words (connected) were found by the endpoint detector and 5 false alarms were also made. Therefore, the new endpoint detector actually found 805 of the 820 digits in the database (98.2 percent). The recognition accuracy on this test set was 90.0 percent, with 711 of the 790 utterances correctly recognized. Since we are using an isolated word recognition system, the recognition errors produced by the 10 double utterances were removed. The 79 errors in the isolated word recognition system were attributed to the following causes:

1. For 10 words, the beginning point included too much of the background noise.

2. For 8 words, the ending point included too much of the background noise.

3. For 4 words, both endpoints were greatly in error.

4. The endpoint detector failed to find the entire word in 12 cases. These all occurred for the digit *six*, where the /IX/ was left out.

5. For the remaining 40 errors, the endpoint detector found the correct endpoints; however, the recognizer was unable to recognize the words.

These errors can be thought of in two ways. Either they were attributable to the endpoint detector, or they were recognizer errors. Types 1, 2, and 3 above are clearly endpoint detector errors. Type 5 is clearly a recognizer error. Type 4 can also be considered a recognizer error, as the template set has tokens for the word *six* without the final /IX/.

If we were to compute a recognition error rate due entirely to the endpoint detector, the errors we would include would be types 1, 2, and 3 above, plus the 5 false alarms and the 15 words missed entirely by the endpoint detector. We are not including the 10 double words found because they clearly were connected words, which could possibly be recognized by a connected word recognition system. Therefore, a total of 42 errors out of a total of 805 utterances (790 utterances plus 15 utterances not found) were due to the endpoint algorithm. This yields an endpoint detector hit rate of 94.8 percent. The corresponding recognizer accuracy, if we eliminate the endpoint error rate, would be 93.3 percent (711 utterances correct out of a possible 763 words). Hence, our recognition results are comparable to those obtained from manual endpoints; however, we suffer a 5-percent error rate in digit detection.

4.5 Alternatives to top-down approach

Several other methods of endpoint detection were examined during our study. Initially, we tried to improve the bottom-up algorithm by first filtering the speech into four bands. Endpoint detection was implemented in each of the four bands and then combined based on a set of rules. The filter bank that we implemented used filters from 100 to 500 Hz, 500 to 1000 Hz, 1000 to 2000 Hz, and 2000 to 3200 Hz, with a small amount of overlapping. This approach yielded results significantly worse than the top-down approach described here. We then included the filter bank approach in the top-down endpoint detector. This also degraded the overall system accuracy. Another technique that was examined, though computationally expensive, was the level-building speech recognition algorithm of Myers and Rabiner.[9] This allowed for an open-ended dynamic time-warp space, and, therefore, the recognizer itself could possibly find the correct endpoints. This approach neither increased nor decreased the accuracy of the endpoint detector.

V. DISCUSSION

Table II shows the results of the recognition experiments run on the different endpoint detection algorithms. The first point to make is that while the new endpoint detector accurately detected 94.8 percent of all words, the old approach only found 68.5 percent. This translates into an recognition error rate component due entirely to nondetection of speech of 5.2 percent and 31.5 percent for the top-down and bottom-up algorithms, respectively.

We also see that the bottom-up algorithm correctly recognized 85.2 percent of the words it detected, while the top-down approach recog-

Table II—Comparison of bottom-up endpoint detector with new top-down endpoint detector

Endpoint Algorithm	Words Detected (Percent)	Recognition Accuracy on Words Detected (Percent)	Overall Recognition Accuracy (Percent)
Bottom-up	68.5	85.2	59.1
Top-down	98.2	90.0	89.4

nized 90.0 percent of the words it detected. If we take into account all errors (endpoints and recognizer) made in processing the 820-word database, the bottom-up endpoint approach led to a total recognition accuracy of 59.1 percent, while the top-down approach had an overall digit recognition accuracy of 89.4 percent. Clearly the top-down endpointing algorithm is superior to the bottom-up approach.

VI. SUMMARY

We have described a new approach to word endpoint detection. We call it a top-down design. Experimental results have been presented indicating that this new approach is able to detect words in highly variable noise environments, as are observed in the telephone network, with much higher accuracy than an earlier implementation of the endpoint detector. The performance of this new technique approaches that of manual endpoint detection—i.e., their recognition accuracies were comparable.

REFERENCES

1. J. G. Wilpon and L. R. Rabiner, "On the Recognition of Isolated Digits From a Large Telephone Customer Population," B.S.T.J., *62*, No. 7, Part 1 (September 1983), pp. 1977–2000.
2. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-29*, No. 4 (August 1981), pp. 777–85.
3. L. R. Rabiner and J. G. Wilpon, "Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition," J. Acoust. Soc. Amer., *66*, No. 3 (September 1979), pp. 663–73.
4. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Recognition—Theory and Selected Applications," IEEE Trans. Commun., *29*, No. 5 (May 1981), pp. 621–59.
5. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-23*, No. 1 (February 1975), pp. 67–72.
6. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 4 (August 1979), pp. 336–49.
7. L. R. Rabiner and J. G. Wilpon, "Speaker Independent, Isolated Word Recognition for a Moderate Size (54 word) Vocabulary," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 6 (December 1979), pp. 583–7.
8. J. G. Wilpon, L. R. Rabiner, and A. F. Bergh, "Speaker Independent Isolated Word Recognition Using a 129 Word Airline Vocabulary," J. Acoust. Soc. Amer., *72*, No. 2 (August 1982), pp. 390–6.
9. 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*, No. 2 (April 1981), pp. 284–297.

AUTHORS

Lawrence R. Rabiner, S.B. and S.M., 1964, Ph.D., 1967 (Electrical Engineering), The Massachusetts Institute of Technology; AT&T Bell Laboratories, 1962—. Presently, Mr. Rabiner is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory* *and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Member, National Academy of Engineering, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

Jay G. Wilpon, B.S., A.B. (cum laude) in Mathematics and Economics, respectively, 1977, Lafayette College, Easton, PA; M.S., 1982 (Electrical Engineering/Computer Science), Stevens Institute of Technology, Hoboken, NJ; AT&T Bell Laboratories, 1977—. Since June 1977 Mr. Wilpon has been with the Acoustics Research Department at AT&T Bell Laboratories, where he is a Member of Technical Staff. His interests include basic research in the field of speech recognition, training algorithms for speaker-dependent and speaker-independent recognizers, and speech endpoint detection.