Speaker Verification by Computer Using Speech Intensity for Temporal Registration

ROBERT C. LUMMIS

Abstract—A technique for automatic speaker verification is described in which voice pitch, low-frequency intensity, and the three lowest formant frequencies, all as functions of time, are the features used to represent an individual utterance. Verification consists of computing these features for a test utterance and comparing them with stored reference versions for the claimed identity. Before the test-versus-reference comparison is effected, the time dimension of the test utterance is warped to optimally register its intensity pattern onto the reference intensity pattern. Performance of the system is measured on a speaker population of moderate size. A variety of comparison formulas and various subsets of the five speech features are evaluated. The system responds either "accept" or "reject" to every utterance; "no decision" is not allowed. Automatic verification based solely upon voice pitch and intensity, both of which can be computed rapidly, yields average error rates below 1 percent.

I. Introduction

In speaker verification, a known cooperative talker is to be distinguished from the set of all other talkers. Verification differs from the more commonly studied problems of talker identification and word identification in several ways, although some of the processing techniques are similar. A key difference is that a verification system need have only two responses: "accept" or "reject" ("no decision" might be included as a third response). Since the number of possible responses is not a function of the population size, response time and accuracy do not deteriorate as the population becomes large. In identification, the number of outcomes equals the number of speakers (or words) the system is supposed to handle, which would have to be very much larger than two to be of commercial value. Because of this, identification is fundamentally more difficult to do, and may not be commercially possible in the forseeable future. The differences between the two problems are discussed in more detail by Li *et al.* [10] and analyzed quantitatively by Doddington [5]–[7].

The first experimental verification system to be described in the literature appears to be that of Li *et al.* They use a spectral representation of the input speech, obtained from a bank of 15 bandpass filters spanning the frequency range 300–4000 Hz. Two stages of adaptive linear threshold elements operate on the rectified and smoothed filter outputs. These elements are "trained" with many utterances of a fixed test phrase by the "true" speaker and by a group of "impostors." The training process results in a set of weights for the various frequency bands and time segments. The weights characterize the speaker. A large number of training and test utterances collected over telephone lines are used to evaluate the system and to study variations in the training procedure. Error rates around 10 percent are reported.

Luck [11] describes a verification system that uses cepstral measurements to characterize two vowels in a standard test phrase. Pitch and word length are also determined. This information is saved for every reference utterance. A test utterance is evaluated by finding the (multidimensional) distance from it to the nearest utterance in the reference set. If this distance is below a threshold value, the utterance is accepted. Error rates with 4 true speakers and 30 impostors average about 8 percent. Luck demonstrates the necessity of collecting reference utterances in a number of separate recording sessions in order to adequately sample the variations in a speaker's voice over time. He also shows that impostors who attempt to mimic the true speaker are unable to significantly improve their ability to do so. Professional mimics were not tested, however, and rehearsal was minimal.

An approach similar to Li's is that of Das and Mohn [1], [2] and Das *et al.* [3]. Their system operates on output signals from a filter bank, but in addition to band energies, it uses pitch and formant information. It basically follows the "Adaline" adaptive procedure, and includes provision for automatic segmentation of the utterance. An error rate of about 1.0 percent is reported. This is quite low, but unfortunately is accompanied by a 10 percent "no decision" rate and is obtained by using 50 training utterances per true speaker, which is inordinately large for commercial application. The performance of this scheme appears to be quite sensitive to successful segmentation, which we believe is a more difficult operation, conceptually, than verification itself. It would seem desirable to replace the segmentation by some more robust method of time normalization, such as that of Doddington [5], [7].

Das *et al.* [4] report further experiments with the same basic system, but with utterances having narrower bandwidth and accompanied by more noise. Error rates around 3.5 percent result. After analysis of an initial set of test utterances, the authors instructed that speaker from the impostor class whom the system rated closest to each true speaker to practice imitating that true speaker. Such practice did not result in utterances substantially closer to those of the true speaker, leading the authors to the conclusion that their system is resistant to mimicking, at least by people inexperienced at it.

Manuscript received September 12, 1972.
The author is with Bell Laboratories, Piscataway, N. J. 08903.

Subjective experiments related to speaker verification are reported by Rosenberg [16]–[18]. He presented human listeners with paired utterances of the phrase "we were away a year ago." The utterances in a pair were sometimes by different speakers and sometimes by the same speaker (in no case were the utterances identical). When intentional mimicry was not involved, listeners made errors at judging same speaker or different speaker 4 percent of the time on the average. When well-rehearsed professional mimics were included among the speakers, their imitations were accepted (i.e., the subjects judged "same") an average of 22 percent of the time.

The system described by Doddington [5]–[7] is fundamentally different from those of Das *et al.* and Li *et al.* He does not use a filter bank, but converts the speech directly to pitch, intensity, and formant frequency values, all sampled 100 times per second. He developed a procedure by which a sample utterance is time registered with a stored reference, representative of the identity claimed, by nonlinear warping of the sample time axis. An accept/reject decision is then made, based on the dissimilarity between sample and reference functions computed with several heuristically chosen formulas. The second formant function plays a key role in the registration procedure. Because it has large clear excursions that are characteristic of the utterance and relatively consistent across speakers and repetitions, it is used as the criterion of good time registration. The warping required to register the second formant is determined; then the other functions are passively warped to the same extent. This time registration method is a major advance over the landmark approaches used by previous workers. Doddington demonstrates that it contributes substantially to the low error rate he obtains, which averages about 1 percent without use of a "no decision" response category. The principal problem with this system, from a commercial standpoint, is the large computing capability required. The formant computation, if done by software, takes about 7 min on a large machine for a 2-s utterance. Fast Fourier transform (FFT) hardware probably would be needed to get a response quick enough for commercial applications.

Doddington reports that his dissimilarity measures based on formant data contribute relatively little to the final accuracy. But the formant computation cannot be omitted because the second formant is required for time registration. The question naturally arises: Can the intensity pattern, although intuitively less well suited to the purpose because of its greater intraspeaker variability, be used in place of the second formant as the criterion of time registration without substantially impairing system performance? We hypothesized that if the intensity were smoothed to eliminate local peaks and valleys, it would be a satisfactory basis for registrations. The intent of the present work was to test this hypothesis by implementing a verification system based on

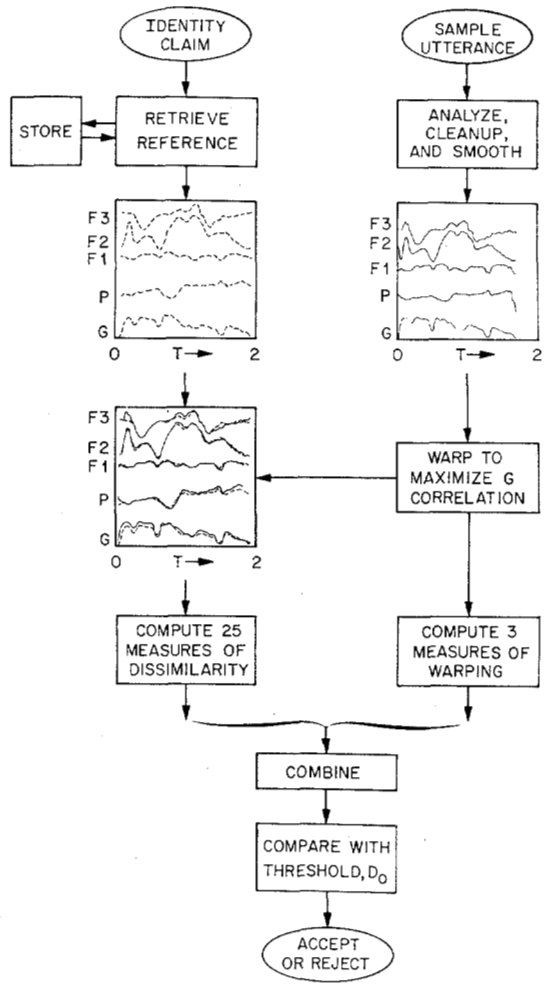


Fig. 1. Block diagram representing the sequence of verification operations.

intensity pattern registration. Formants are still computed so that their relative contribution to overall accuracy can be assessed, but they are no longer indispensable. The overall scheme is similar to Doddington's, the principal differences being: 1) different dissimilarity formulas are used; 2) time registration is based on the intensity pattern instead of the second formant (Doddington's registration program is used verbatim); 3) the speech functions are smoothed by low-pass filtering; and 4) reference utterances are constructed differently.

Our scheme is summarized by the block diagram in Fig. 1. There are two inputs to the system, shown at the top of the diagram. An identity claim is accepted at the left. Reference information, including prototype speech functions and variability data, is fetched from storage for the identity claimed. A sample utterance is accepted at the right. The sample is "analyzed" to obtain the five speech functions: gain,[1] pitch, and three formant frequencies, which are smoothed by a 16-Hz low-pass filter. The time axis of all five functions is then warped

[1] The "gain" function is synonomous with the intensity pattern. It is the 600-Hz low-passed speech intensity smoothed with a 16-Hz low-pass filter.

to produce temporal registration of the sample gain function and the prototype gain function. The overall dissimilarity between sample and prototype functions is measured, with due allowance for the variability of the claimed speaker. If the overall dissimilarity exceeds a threshold value, the system response is "reject;" otherwise, it is "accept." The significant steps are described in greater detail in what follows.

This system was implemented on a large general-purpose computer (Honeywell-635) [12]. Subsequent to the work described in this paper, another version of the system, one that does not compute formants, was implemented on a small laboratory machine (DDP-516). This version provides a hands-on interactive facility with graphical displays that is useful for studying proposed system improvements [13].

II. Speech Analysis

The system does not deal with speech in terms of its acoustic waveform directly, but rather in terms of "speech functions," or "control functions," which are computed in the "Analyze" box in Fig. 1. These are functions of time that describe the speech parametrically, and are called control functions because a complete set of them would suffice to *control* a speech synthesizer to produce a version of the utterance perceptually similar to the original. The control functions we use are five in number: pitch period P, gain G, and first, second, and third formant frequencies $F1$, $F2$, and $F3$. These functions are computed by the FFT cepstral method of Schafer and Rabiner [20].

In outline, this method is as follows. The speech is low-pass filtered at 4 kHz and sampled with 11-bit precision at 10 kHz. A sliding FFT of the speech is computed using a Hamming window, whose duration is approximately four times the pitch period. This yields a transform sampled every 10 Hz, spanning the range 0–5000 Hz. The time window is moved 100 samples, or 10 ms, for successive FFT computations. The sum of squares of the first 60 spectrum samples is taken as a "gain" function sample G; thus, the gain is the 600-Hz low-pass speech intensity. The log magnitude of the transform is subjected to a second FFT computation, yielding the speech cepstrum. A peak-picking routine applied to the high-time end of the cepstrum locates the pitch peak (if the speech was voiced), and produces from it one sample of the pitch period control function P. The cepstrum is low-time filtered at 4 ms to remove the pitch peak, then inverse transformed to yield a smoothed spectral envelope of the speech. Peak-picking logic, operating on this envelope, generates one sample for each of the $F1$, $F2$, and $F3$ functions.

These five functions are adequate for representing most voiced speech sounds. Additional parameters would be needed for unvoiced speech and for nasals. We use only the all-voiced nonnasal test utterance, "We were away a year ago," so that the five control functions described above are sufficient.

There are certain difficulties in the analysis operation. For example, it is not always possible to produce a meaningful sample of the pitch and formant tracks at every sampling point in the utterance. The pitch period, for example, cannot be measured if voicing fails or when the speech level drops to zero, such as it does for the "G" in "ago." Logic is included in the analysis algorithms to detect such problem points and generate a special code value instead of a defective sample value. All of the programming that subsequently deals with the control functions must include provision for special handling of these "blank" samples. This requirement accounts for a considerable amount of complexity.

A problem that is treated differently is a type of irregularity, characteristic of some speakers, in which the pitch track includes an abrupt large jump followed by an equal jump in the opposite direction after a few samples. An example is shown in Fig. 2(a). If these jumps were consistent, they could be considered characteristic of the speaker and used for verification. Unfortunately, they are not at all consistent, so the pitch tracks are edited to eliminate them. If the size of each jump of the pair is an octave (within 3 Hz), both jumps are eliminated to yield a smooth pitch track. If both are not octave jumps, but the duration of the jump is short, it is eliminated by removing the offending samples and filling in the gap by linear interpolation. Pitch editing of one of these two types was applied to 6 percent of the utterances in our test population. In other cases of jumps, where the proper corrective action is not obvious and unambiguous, the special code value is substituted to indicate a blank sample.

Certain difficulties in pitch and formant extraction that are prevalent near the beginning and end of utterances are handled by converting sample values at both ends to blanks. Enough blanks are used to eliminate sharp jumps near either end of any of the control functions.

We believe that in future work, many of the pitch-extraction difficulties can be circumvented by measuring pitch only in high-level well-voiced portions of the test utterance. Further, it may be sufficient merely to determine the average of such easily measured values over the whole utterance.

After the "cleanup" operation described above, the control functions are smoothed with a low-pass filter designed by the method of Rabiner *et al.* [15]. This design provides an exactly linear phase characteristic, and minimizes peak out-of-band transmission. Fig. 2(a) and (b) show the same utterance before and after the cleanup and smoothing.

Smoothing of the gain function is essential if it is to be used for time registration. The other functions are smoothed for symmetry and because their higher frequency variations are presumed to be unimportant for

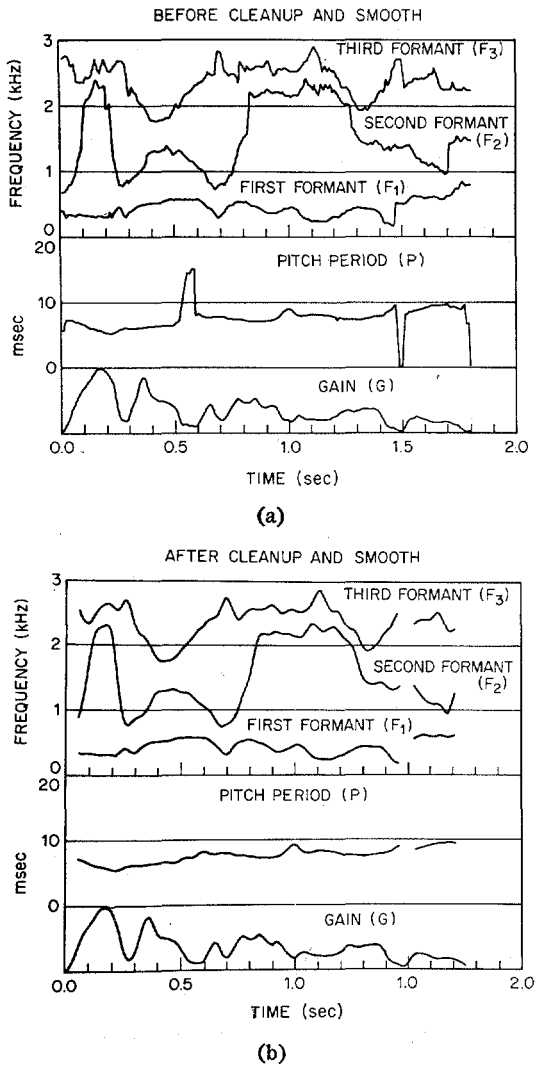


Fig. 2. Plots of a typical set of control functions used for verification. The speaker said, "We were away a year ago." The utterance shown illustrates special handling of certain problems. Part (a) shows the functions before cleanup and smoothing. There is one interval in which exact (as defined in the text) pitch period doubling occurs and one interval in which pitch tracking fails. Part (b) shows the same utterance after cleanup and smoothing: the curves have all been low-pass filtered at 16 Hz. The double pitch period interval has been "corrected" by exactly halving the sample values involved, and the interval of no pitch tracking has been deleted altogether from the pitch and formant curves. One or the other of these two kinds of special handling was required for only a small minority of the test utterances.

speaker verification. The question of what frequency ranges in the control functions are useful for speaker verification is an important one and has not been studied. If only very low frequency components would suffice, lower sampling rates could be used with attendant advantages of reduced storage and quicker processing. The smoothing filter used here was 16-Hz low pass. This was chosen because studies of the subjective quality of speech resynthesized from (low-pass) filtered control functions indicate that this cutoff frequency can be used without a noticeable degradation in speech quality [19].

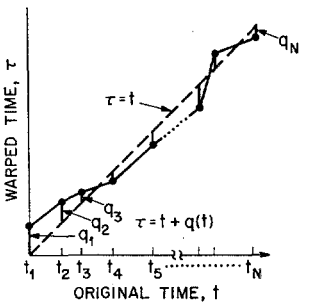


Fig. 3. Plot of a typical warping function that is used to specify the relation between warped and unwarped time.

III. Time Warping

Before a sample utterance can be evaluated for verification, it must be brought into time registration with the prototype utterance with which it is to be compared. This is accomplished by the method of time warping described by Doddington. In fact, his program is used verbatim. Briefly, an additive "warping function" $q(t)$ is determined, which maps sample points of a time function $x(t)$ onto a warped time axis τ. The warping function is constructed to be piecewise linear and continuous so that it is specified completely by the coordinates of its breakpoints. Thus,

$$\tau(t) = t + q(t)$$

and

$$q(t) = q(t, t_i, q_i), \qquad i = 1, N.$$

We have limited the number of breakpoints N to 10, so that, at most, 20 parameters are required to specify the warping. The relationship between warped and unwarped time is illustrated in Fig. 3. Following the method of iterative steepest ascent, each of the $2N$ parameters is incremented by a small amount that is proportional to the partial derivative with respect to the parameter in question of the coefficient of correlation ρ_{xy} between the warped function $x(\tau)$ and the function with which it is to be registered $y(t)$. The coefficient of correlation is defined as

$$\rho_{xy}(t_i, q_i) = \frac{\overline{x(\tau) \cdot y(t)}}{\left[\overline{x^2(\tau)} \cdot \overline{y^2(t)}\right]^{1/2}}.$$

The overbar in this formula means time average computed over that range of t and τ within which both $x(\tau)$ and $y(t)$ are well defined and nonzero. Empirically determined constraints incorporated in the process prevent too large an increment on any single iteration, and prevent the warping function from becoming too severe. The latter constraint is needed, for example, to avoid shrinking the function to zero length. Zero length would yield the globally optimal correlation coefficient of 1.0, but would not be a meaningful time registration. The number of breakpoints is automatically decreased when two get very close together, and increased if two con-

secutive ones get too far apart. Iterations are continued until one of the following conditions is satisfied after iteration i:

$$1)\ i = i_{\max} = 35$$
$$2)\ \rho_{xy,i} \geq \rho_{\max} = 0.999$$
$$3)\ \rho_{xy,i} \geq 0.98$$

and

$$\left| \rho_{xy,i} - \rho_{xy,i-1} \right| / (1 - \rho_{xy,i-1}) \leq 0.02$$

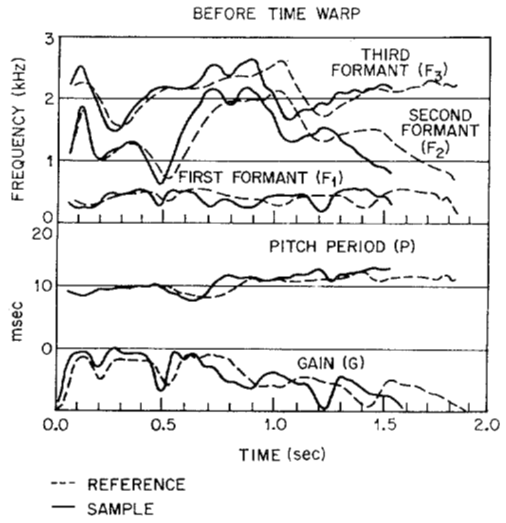
and

$$\left| \rho_{xy,i-1} - \rho_{xy,i-2} \right| / (1 - \rho_{xy,i-2}) \leq 0.02.$$

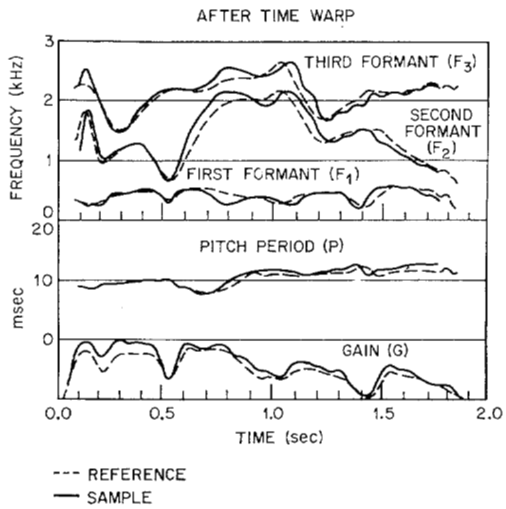
The third condition means that after the correlation exceeds 0.98, the absolute value of the fractional change in the deviation of the correlation from 1.0 must be below the criterion value of 0.02 on two consecutive iterations.

The warping procedure described here is not applied to each of the control functions separately. It is only applied to the gain. The warping function that results is then applied directly to the remaining control functions without further iteration, whether or not they are optimally registered thereby. Fig. 4 illustrates the relationship between a sample utterance and the prototype for the same speaker before and after warping.

Registration of a new utterance with a prototype required, on the average, 19 iterations for a "true" speaker and 33 iterations for an impostor. The iteration process was seldom terminated by criterion 1) for a true speaker, but was usually so terminated when the speaker was an impostor.



(a)



(b)

Fig. 4. Comparison of the five control functions of a sample utterance with prototype functions for the same speaker, (a) before and (b) after the time registration operation. The time axis of the sample gain function is warped by an iterative algorithm that seeks to maximize the correlation coefficient between sample and prototype gains. The other four functions are passively time warped to the same extent.

IV. Prototype Utterances

To characterize each speaker, a set of so-called *specimen* utterances are used. In a working system, these would be collected when the speaker's identity was not in question. They are analogous to the specimen signature collected from a depositor when he opens a signature savings account, and they serve as the standard against which future samples are compared. In our system, the separate specimen utterances are not all preserved, but instead, a single *prototype* utterance is generated and stored as part of the reference information.

The prototype is derived from the set of specimens by the following algorithm.

1) Stretch or contract linearly all specimens to a standard length of 1.9 s. This is done by warping each with a linear warping function.

2) Average the specimens at each sample point, producing the first "trial prototype."

3) Register each specimen to the trial prototype by means of the iterative warping procedure described in Section III.

4) Average the registered specimens to form a new trial prototype.

5) Determine whether or not the new prototype is significantly different from the previous one. The criterion for significantly different is that at least 3 percent of the values differ by at least 3 percent. If it is not significantly different, stop. If it is, repeat from step 3) above.

The final prototype produced at step 5) is stored for subsequent use in verification. In addition, a large number of variances are computed and stored, which are explained in Section V. These quantities represent the scatter among the set of registered specimens, measured in terms of the various distance formulas that will be used for verification of a sample utterance. We refer

to the five prototype functions and the set of variances jointly as the "reference" information.

V. Distance Computation

The degree to which a given utterance differs from a prototype, after time registration, is the quantity upon which the verify/reject decision is based. The formulas used to measure this difference are so constituted that a value of zero results if an utterance is identical to the prototype to which it is compared, and a positive value results if the utterance and prototype differ. Larger (positive) values result from greater differences. Because of these properties, the measurements are called distances. We compute 28 separate distances, which are then combined to yield a measure of overall dissimilarity. The various distance measures now will be defined. For expository purposes, we distinguish two categories of distances—those based on short segments of the utterance treated separately, and those computed over the whole of the utterance.

For the first type, the 2-s time frame containing the control functions is divided into 20 equal segments, labeled $i=1, 2, \cdots, 20$. Because the control functions are sampled at 100 Hz, each segment contains ten sample points. Provided the time registration was successful, each segment of the sample utterance will contain the same linguistic material as the corresponding segment of the prototype. Of course, the sample and the prototype will never be identical, even if the speaker's identity claim is true. The extent of the difference between them depends, for a given speaker, upon three things: the segment in question, the control function in question, and the formula being used to measure the difference. Since there are 20 segments, 5 control functions, and 4 segmental distance formulas, the variability of this difference for a given speaker may be represented by a $20\times5\times4$ matrix. This variability matrix contains values that are averages, over the set of specimen utterances, of certain quantities used in the distance formulas. This is explained more precisely below.

The four segmental distance formulas that are applied to each control function at verification time are

$$d_1 = \frac{1}{20}\sum_{i=1}^{20}(a_{x,i} - a_{p,i})^2/{\sigma_{a,i}}^2$$

$$d_2 = \frac{1}{20}\sum_{i=1}^{20}(b_{x,i} - b_{p,i})^2/{\sigma_{b,i}}^2$$

$$d_3 = \frac{1}{20}\sum_{i=1}^{20}(c_{x,i} - c_{p,i})^2/{\sigma_{c,i}}^2$$

$$d_4 = \frac{1}{20}\sum_{i=1}^{20}(1 - \rho_{xp,i})^2/{\sigma_{\rho,i}}^2.$$

Suffixes x and p refer to unknown (i.e., test) and prototype utterances, respectively, and subscript i designates a time segment ($i=1, 2, \cdots, 20$). Symbols a, b, and c are the first three coefficients of an orthogonal polynomial representation of the control function samples comprising a particular time segment. In our case, with ten points per segment, the first three orthogonal polynomials are [9]

$$P_{0j} = 1$$
$$P_{1j} = j - 5.5$$
$$P_{2j} = j^2 - 11j + 22.$$

Thus, if the control function samples for an unknown utterance within the segment being measured are x_j ($j=1, 2, \cdots, 10$), then

$$a = \left(\sum_{j=1}^{10} x_j\right) \Big/ 10$$

$$b = \left(\sum_{j=1}^{10} x_j P_{1j}\right) \Big/ \sum_{i=1}^{10} {P_{1j}}^2$$

$$c = \left(\sum_{j=1}^{10} x_j P_{2j}\right) \Big/ \sum_{j=1}^{10} {P_{2j}}^2.$$

In the formula for d_4, $\rho_{xp,i}$ is the coefficient of correlation between an unknown and a prototype control function in segment i.

$$\rho_{xp,i} = \frac{\sum_{j=1}^{10} x_{ji}\cdot p_{ji}}{\left[\sum_{j=1}^{10} {x_{ji}}^2 \cdot \sum_{j=1}^{10} {p_{ji}}^2\right]^{1/2}}$$

where x_{ji} and p_{ji} are the sample points ($j=1, 2, \cdots, 10$) of the unknown and prototype control functions within segment i.

The quantity in the denominator of each of the distance formulas is the measure of variability mentioned previously. It is the average, over the set of registered specimen utterances, of the squared difference indicated in the numerator. For example, the denominator for d_1 is

$${\sigma_{a,i}}^2 = \frac{1}{n_s - 1}\sum_{s=1}^{n_s}(a_{s,i} - a_{p,i})^2$$

where subscript s designates a specimen utterance, and n_s is the number of specimens. In our study, $n_s=10$. Since the sample values of the prototype are the averages of the specimen sample values, this formula is close to the definition of a variance; hence we use the symbol σ^2 for this measure of variability. This completes the explanation of the segmental distances that are used.

Additional distances are computed that are defined over the whole utterance instead of segment-by-segment. One of these is a fifth distance measured on each of the five control functions:

$$d_5 = (1 - \rho_{xp})^2/\sigma_p^2.$$

In this formula, ρ_{xp} is the correlation coefficient between unknown and prototype functions measured over the whole length of the utterance, and σ_p^2 is the average over the specimen set of $(1-\rho_{sp})^2$. The remaining three distances depend on the degree of warping required for time registration. The first three orthogonal polynomial coefficients are computed for the warping function q as a function of t. These coefficients are called a_q, b_q, and c_q. The second coefficient, representative of the average slope of q versus t, indicates the amount of overall slowdown (if positive) or speedup (if negative) required to be applied to the sample to achieve registration with the reference. The first "warping distance" is defined as

$$d_{q1} = (b_q - \bar{b}_q)^2$$

where $\bar{b}_q$ is the average value of b_q over the set of registered specimen utterances. The third orthogonal polynomial coefficient is used to define a second warping distance:

$$d_{q2} = c_q^2.$$

Finally, the variation of q as a function of t that remains after the first three orthogonal polynomials are subtracted is termed r_q. Its mean-square value is the third warping distance:

$$d_{q3} = \overline{r_q^2}.$$

Thus, a total of 28 distances are measured that characterize the dissimilarity between an unknown and a prototype utterance: $d_1, d_2, \cdots, d_5$ for each of the five control functions, and three distances obtained from the warping function. All 28 distances, or any desired subset of the 28, are combined together to form a single measure of dissimilarity, referred to as the overall dissimilarity. Although clearly suboptimal, this combination is done by simply adding them together after first normalizing each by dividing it by its average over the specimen set. This normalization is necessary to make all measures comparable, and has the advantage that it tends to equalize their variance.

VI. Test Utterances

The system was evaluated by processing a group of utterances with it and noting the percentage of correct decisions. The utterances used were those collected by Doddington and used by him to measure the performance of his system. Forty-one speakers are included. They are all males, and none has any pronounced voice peculiarity. Eight are designated "customers," 32 are "casual impostors," and the last is an identical twin brother of one of the customers. A "casual impostor" does not attempt to imitate anyone; he speaks in his own natural voice. He is an impostor in the sense that his utterance is presented to the system with an identity claim of one of the customers. The number of utterances available is shown in Table I.

TABLE I
Test Utterance Population

Speaker Class	Number of Speakers	Utterances per Speaker	Total Utterances
Customer	8	15	120
Impostor	32	1	32
Twin	1	11	11
Total	41	—	163

The recordings were made in a sound booth using high-quality equipment. At least two days elapsed between consecutive utterances, so that normal speaker variations could be represented adequately. An effort was made to prevent any speaker from ever hearing the test phrase spoken by anyone except himself. The utterances were low passed at 4-kHz, digitized at 10-kHz sampling rate with 11-bit precision, and converted to pitch, gain, and formant functions, which were cleaned up and smoothed, as explained in Section II.

VII. System Performance

For each customer, three prototypes, designated "A," "B," and "C," were constructed from different subsets of 10 of the 15 available utterances. The utterances used to construct a given prototype are called "specimen" utterances with respect to that prototype. The remaining utterances by the same speaker are called "true" utterances, and the 32 utterances by impostors are called "false" utterances. An error rate is determined for each prototype by the following procedure. The overall distance (see Section V) from the prototype to each of the 5 true and 32 false utterances is computed from whatever subset of the 28 distances is desired. The acceptance/rejection threshold is determined from this population of 37 overall distances by the method explained below, and each utterance is then classified as "accepted" or "rejected," depending upon whether its overall distance is less than or greater than the threshold value, respectively.

There are two kinds of errors that the system can commit. It can reject a "true" utterance, and it can accept a "false" utterance. The relative frequency of each error type is controlled by the value chosen as threshold. If the threshold is high, few true utterances will be rejected, but many false ones will be accepted. Conversely, a low threshold will produce a preponderance of false rejections. A compromise is necessary. In a commercial application, the threshold would be set to minimize some measure of overall cost that took into account the relative undesirability of the two kinds of error.

For our purpose, which is to evaluate the system and to compare the efficacy of various distances, we use the *a posteriori* equal-error threshold, determined as follows. Consider the five distances for the true utterances to be arranged in order of descending value, and a plot to be

prepared showing the proportion (on the ordinate) of distances that are less than a specified value (on the abscissa). This will form a descending staircase with horizontal segments at ordinate values 1.0, 0.8, 0.6, 0.4, and 0.2, and vertical segments at abscissa positions representing the five true distances. Similarly, consider the 32 false distances arranged in increasing order and plotted as an ascending staircase, showing the proportion of distances less than the abscissa value. The abscissa value at the intersection of these two plots is taken as the acceptance/rejection threshold. The ordinate at this intersection is the equal-error rate.

A comparison of such equal-error rates, for various subsets of the distances, is given in Table II. It should be understood that no matter what distances are used, time registration is first carried out with the gain function. The values in the table are averages of 24 equal-error rates (3 prototypes per customer, 8 customers), expressed as percent and rounded to two significant digits. The quantized nature of the calculation restricts these averages to certain values. Below 0.83 percent, for example, they must be integer multiples of 0.1302 percent. Each entry is the error rate that results when the distance formula indicated at the left is applied to the control function in the column heading. The symbol $\sum_{i=1}^{5} d_i$ indicates a distance that is the sum of the five individual distance formulas. The entries with range bars in the bottom section are the average error rates that result when the overall distance is the sum over the five distance formulas and over the group of control functions indicated by the bar. For example, when the overall distance is taken as the sum of all five formulas applied to the pitch function, plus the sum of all five formulas applied to the gain function, plus the sum of the first three formulas applied to the warping function (d_4 and d_5 are undefined for warping), the average equal-error rate that results is 1.2 percent.

Because of the small sample size and the way the samples were rotated, there is no good way to specify the uncertainty of the values in Table II. We believe, however, that at least the main differences that show up are real. Evidence for this is that the error rates for most of the individual speakers show the same effects as the averages in Table II. For example, the error rates based on all five distance formulas for pitch are lower than the corresponding rates for any one of the formants for seven of the eight customer speakers. A speaker-by-speaker comparison of the error rate for gain with the rate for any one of the formants yields the same result. Further, four out of five of the separate distance formulas give a lower error rate for pitch and for gain than for any formant. This kind of pattern would be very unlikely if the differences in the table were statistically insignificant.

It is not clear whether or not the smaller effects in Table II are significant. That is, the error rate for all measures combined (shown as 1.0 percent) may, in fact, not be lower than the error rate for pitch, gain, and warping (shown as 1.2 percent). And this, in turn, may not be significantly different from the rate for pitch and gain alone (0.5 percent). The statistical significance of these lower error rates is suspect because the total test population contains only 152 independent utterances.

TABLE II

Equal-Error Rates (in Percent) for Various Combinations of Distance Measures

DISTANCE	CONTROL FUNCTION					
	PITCH	GAIN	WARP	F 1	F 2	F 3
d_1	4.3	13	25	12	12	11
d_2	14	6.8	38	19	17	19
d_3	23	12	10	22	24	23
d_4	10	4.7		17	15	16
d_5	7.2	5.5		15	12	13
	4.1	1.7	16	11	13	11
$\sum_{i=1}^{5} d_i$	0.5			9.7		
	1.2				10	
				8.4		
	1.0					

The test population on which the error rates in Table II are based does not include the utterances of the twin speaker. All 11 of the twin utterances were correctly rejected by the system. It is interesting to note that in an experiment in which listeners verified speakers' identity auditorily, the same twin impostor utterances were almost always accepted as those of the true speaker. This difference is attributable to the fact that listeners paid too much attention to voice quality, which was very similar in the twins, and were willing to overlook substantial difference in timing and intensity patterns [18].

VIII. Conclusions

Three conclusions can be drawn from these figures. First, because the best error rates in the table are quite low—on the order of 1 percent—we conclude that the basic approach is a viable one. In other words, verification is a practical possibility. Absolutely secure automatic verification (i.e., an error rate very much less than 1 percent) appears unobtainable, if for no other reason than that a "true" speaker will occasionally render a defective utterance. But a system that performs at about the 99 percent-correct level should find many commercial applications. This conclusion agrees with previous studies (e.g., Doddington [5]–[7]).

The second conclusion is that in spite of the fact that

time warping was performed in such a way as to optimally register the gain function, the use of only the gain and pitch functions for verification provides very good performance. In fact, distances based on formant frequencies contribute very little toward discriminating the casual impostor. This may be explained (or at least rationalized) by noting that formant values are constrained, to a much greater degree than are gain and pitch, by the words being pronounced. For that reason, personal idiosyncrasies probably can manifest themselves most readily in the pitch and gain function so that distances based on those functions would be the most suitable for verification.

Finding that formants are not necessary for distance computation, even when the registration is based on gain, is of considerable importance, and answers the principal question that motivated this work. Since our implementation does not need formants for time registration, it means that formant computation can be entirely eliminated from the process.[2] Formant computation is quite time consuming (about 7 min of computation for a 2-s utterance on our large general-purpose computer). Its elimination speeds up the process by more than an order of magnitude. An alternate means of pitch computation would be necessary, since we obtain pitch as a byproduct of formant computation, but suitable methods are available (e.g., Gold and Rabiner [8]).

Our third conclusion is that it is possible to obtain particularly low error rates with high computing efficiency using a small set of judiciously chosen individual measures. Many combinations of individual measures were tried and, although not indicated in Table II, some yielded very good performance. A combination of particular interest is $d_{1\,\text{pitch}}+(d_2+d_3+d_5)_{\text{gain}}$. It is of interest because it is especially easy to measure, and yields the very low error rate of 0.65 percent. Except for the differential weighting of various time segments, as described in Section V, $d_{1\,\text{pitch}}$ is simply the difference between average pitch of the test and prototype utterances. That appears to be sufficient information about the pitch. The principal effect of the segmental weighting is to eliminate those segments in which unusual pitch values occur. In nearly all cases, these are the segments having low gain. Thus, it may be adequate in future implementations to measure pitch only in high-gain portions of the utterances (where, incidentally, it is most easily measureable), and then preserve only the average value so obtained and one measure of its variability. This will eliminate the need for almost all the pitch reference information that is currently used (200 words for pitch values and 80 words for pitch variances). A few additional simplifications arise from the use of only d_2, d_3, and d_5 measures for gain. (The average-sensitive measure d_1 does not have as great a discriminating power as one might expect because all gain functions are normalized to have a predetermined peak value. This normalization is necessary to allow for reasonable variations in microphone gain, but it obviously eliminates useful information from the utterance about overall voice level of the talker.) The variances for $d_{2\,\text{gain}}$ and $d_{3\,\text{gain}}$ are likely to vary from segment to segment in a similar way. If only one set of variances is preserved and used for both d_2 and d_3 (none is required in the computation of d_5), the reference storage requirement for gain is reduced to 220 words. The total storage, if $d_{1\,\text{pitch}}+(d_2+d_3+d_5)_{\text{gain}}$ is used as the overall distance measure, thus would be only 222 words. This compares with 1400 words of storage needed for the prototype information used in the current study.

It is important to note that the low error rate found for the measure $d_{1\,\text{pitch}}+(d_2+d_3+d_5)_{\text{gain}}$ does not have much statistical significance because of the small size of our test population. Even if the true error rate were somewhat higher, however, it would still be a valuable measure because of its computational convenience. A test of our system with a much larger population of utterances seems desirable.

[2] A separate study in which professional mimics attempted to imitate customer voices indicates that in a special situation where the error rate is higher, the information provided by formants is of value. One might imagine a system in which formant computation and comparison would be used in a second stage of verification when the gravity of the transaction warranted extra computation time [14].

References

[1] S. K. Das and W. S. Mohn, "Pattern recognition in speaker verification," in *1969 Fall Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 35. Montvale, N. J.: AFIPS Press, 1969, pp. 721–732.

[2] ——, "A scheme for speech processing in automatic speaker verification," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 32–34, Mar. 1971.

[3] S. K. Das, W. S. Mohn, and S. L. Saleeby, "Speaker verification experiments," *J. Acoust. Soc. Amer.*, vol. 49, p. 138(A), 1971.

[4] S. K. Das, W. S. Mohn, S. S. Willett, and W. D. Chapman, "Two speaker verification experiments," in *Proc. IEEE/AFCRL 1972 Conf. Speech Communication and Processing*, pp. 275–278.

[5] G. R. Doddington, "A computer method of speaker verification," Ph.D. dissertation, Dep. Elec. Eng., Univ. Wisconsin, Madison, 1970.

[6] ——, "A method of speaker verification," *J. Acoust. Soc. Amer.*, vol. 49, p. 139(A), 1971.

[7] ——, "A method of speaker verification using nonlinear time registration," submitted to *J. Acoust. Soc. Amer.*

[8] B. Gold and L. R. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time domain," *J. Acoust. Soc. Amer.*, vol. 46, pp. 442–448, 1969.

[9] N. L. Johnson and F. C. Leone, *Statistics and Experimental Design in Engineering and the Applied Sciences*, vol. 1. New York: Wiley, 1964, pp. 426–432.

[10] K.-P. Li, J. E. Dammann, and W. D. Chapman, "Experimental studies in speaker verification, using an adaptive system," *J. Acoust. Soc. Amer.*, vol. 40, pp. 966–978, 1966.

[11] J. E. Luck, "Automatic speaker verification using cepstral measurements," *J. Acoust. Soc. Amer.*, vol. 46, pp. 1026–1032, 1969.

[12] R. C. Lummis, "Real-time technique for speaker verification by computer," *J. Acoust. Soc. Amer.*, vol. 50, p. 106(A), 1971.

[13] ——, "Implementation of an on-line speaker verification scheme," *J. Acoust. Soc. Amer.*, vol. 52, p. 181(A), 1972.

[14] R. C. Lummis and A. E. Rosenberg, "Test of an automatic speaker verification method with intensively trained professional mimics," *J. Acoust. Soc. Amer.*, vol. 51, pp. 131(A)–132(A), 1972.

[15] L. R. Rabiner, B. Gold, and C. A. McGonegal, "An approach

to the approximation problem for nonrecursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 83–106, June 1970.
[16] A. E. Rosenberg, "Listener performance in a speaker verification, task," *J. Acoust. Soc. Amer.*, vol. 50, p. 106(A), 1971.
[17] ——, "Listener performance in a speaker-verification task with deliberate impostors," *J. Acoust. Soc. Amer.*, vol. 51, p. 132(A), 1972.
[18] ——, "Listener performance in speaker verification tasks," in *Proc. IEEE/AFCRL 1972 Conf. Speech Communication and Processing*, pp. 283–286.
[19] A. E. Rosenberg, R. W. Schafer, and L. R. Rabiner, "Effects of smoothing and quantizing the parameters of formant-coded voiced speech," *J. Acoust. Soc. Amer.*, vol. 50, pp. 1532–1538, 1971.
[20] R. W. Schafer and L. R. Rabiner, "System for automatic formant analysis of voiced speech," *J. Acoust. Soc. Amer.*, vol. 47, pp. 634–648, 1970.

A Descriptive Technique for Automatic Speech Recognition

RENATO DE MORI

Abstract—**A technique is introduced that analyzes the time evolutions of some parameters of the speech waveform. Suitable algorithms provide a description of these evolutions when a word is pronounced. Descriptions can be seen as the phrases of a language produced by a generative grammar. Recognition is performed by parsing the descriptions. Some experimental results are reported.**

I. Introduction

The purpose of this paper is to introduce a descriptive technique for automatic speech recognition. The basic idea of this technique is that of analyzing and describing the time evolutions of some parameters obtained from the speech waveform. The parameters used are the gravity centers of the zero-crossing interval distributions obtained at the output of two filters in accordance with a technique introduced by Sakai *et al.* [1].

The validity and the limits of zero crossings as elements bearing useful information for speech recognition have been evidenced by many theoretical and experimental works [2]–[15].

The parameters mentioned have been found useful for obtaining a concise and meaningful graphical representation of a spoken word [15]. The recognition process acts on these graphs with suitable algorithms generating a description of the local aspects of the graph.

The stationary and the nonstationary segments of the speech waveform are singled out, and a list is produced containing a qualitative description of the nature of those segments and the values of the most important attributes (for example, the duration of each segment). Then, the local aspect descriptions are composed leading to a global aspect description that takes into account the relations between properties of each segment of the speech waveform. Recognition is performed by analyzing the global aspect descriptions with a set of acceptors, with each one having to recognize just one word.

Local aspect descriptions can be also seen as terminal syntactic elements of a generative grammar that generates all the descriptions obtained by the pronunciation of a word belonging to the limited vocabulary the machine must recognize. Thus, the recognition process is a way of parsing the local aspect description.

The parsing procedure mentioned above has been employed in the recognition of the ten spoken digits. A recognition rate of 98 percent for four male speakers has been reached with an acceptable computation time. The vocabulary can be extended by adding new acceptors. It is easy to modify an acceptor, which is programmed by a punched tape, if it initially does not recognize a word.

II. Graphical Representation of a Spoken Word

A. The Electroacoustic Chain

Sounds, converted by the microphone to electrical signals, enter a preprocessing unit consisting of an amplifier and an envelope detector. Next, the signal is delivered to two filters connected in parallel: one is a low-pass filter (LPF) with a 1100-Hz cutoff frequency, the other is a high-pass filter (HPF) with a 500-Hz cutoff frequency. The outputs of these filters are interfaced with a DDP-516 Honeywell computed by a multiplexer and a 10-bit A/D converter. The output of the envelope detector is compared with an adjustable fixed voltage and, if higher, it enables the computer to detect the zero crossings of the incoming signals (Fig. 1) and to process them up to the printing of the recognized word.

B. Analysis of Zero Crossings

A careful analysis of the sequences of zero-crossing intervals from many words pronounced by several male

Manuscript received May 17, 1972. This work was supported by the Consiglio Nazionale delle Ricerche of Italy and was performed at the Centro di Elaborazione Numerale dei Segnali.

The author is with the Istituto di Elettrotecnica Politecnico di Torino, Turin, Italy, and the Centro di Eleborazione Numerale dei Segnali, Turin, Italy.