SPEECH RESEARCH DIRECTIONS

Bishnu S. Atal *is head of the Acoustics Research Department and* ***Lawrence R. Rabiner*** *is head of the Speech Research Department at AT&T Bell Laboratories in Murray Hill, New Jersey. Mr. Atal's research interests have centered on speech communication and new methods for analyzing and synthesizing speech signals based on linear prediction. He has a B.Sc. in physics from the University of Lucknow, India; a diploma in electrical communications engineering from the Indian Institute of Science, Bangalore; and a Ph.D. in electrical engineering from the Polytechnic Institute of Brooklyn, New York. Mr. Rabiner engages in research on speech communications and digital signal processing techniques. He has an S.B., S.M., and Ph.D. in electrical engineering from Massachusetts Institute of Technology.*

This paper presents an overview of the current activities in speech research. We will discuss the state of the art in speech coding, text-to-speech synthesis, speech recognition, and speaker recognition. In the speech coding area, current algorithms perform well at bit rates down to 9.6 kb/s, and the research is directed at bringing the rate for high-quality speech coding down to 2.4 kb/s. In text-to-speech synthesis, what we currently are able to produce is very intelligible but not yet completely natural. Current research aims at providing higher quality and intelligibility to the synthetic speech that these systems produce. Finally, today's systems for speech and speaker recognition provide excellent performance on limited tasks; i.e., limited vocabulary, modest syntax, small talker populations, constrained inputs, etc. Current research is directed at solving the problem of continuous speech recognition for large vocabularies, and at verifying talkers' identities from a limited amount of spoken text.

Introduction

Although the field of speech research is broad and encompasses several diverse application areas, we will be concerned here only with speech coding, synthesis, and recognition. We will review the status (at AT&T Bell Laboratories) of these important application areas, discuss relevant issues that are limiting performance, and identify the directions in which we see the research heading.

Speech coding involves communication between people and, therefore, deals with techniques of speech transmission over the telephone system. Of prime concern here are methods for reducing the required bandwidth (or, equivalently, the bit rate) for transmitting speech.

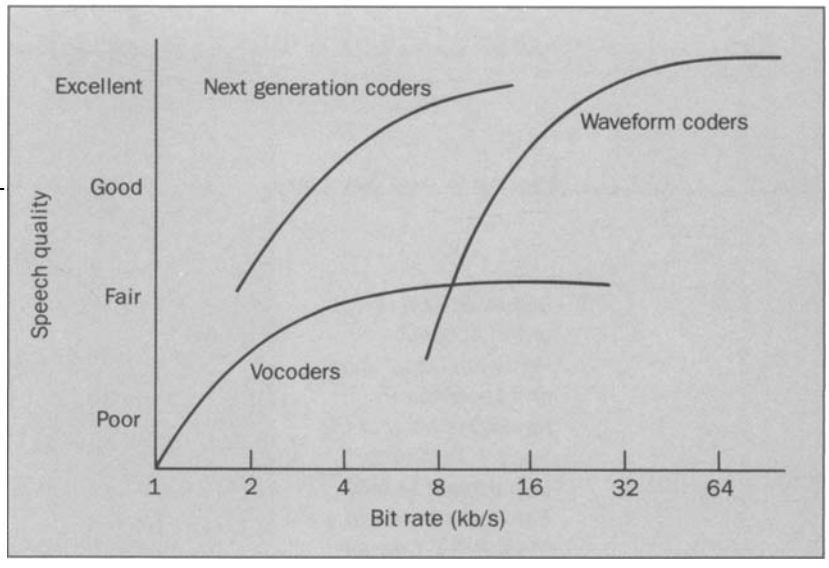

Figure 1. Speech quality versus bit rate for different types of coders.

Speech synthesis, or computer voice response as it is often called, involves machines talking to people. Although systems as simple as announcement machines fall into this area, we will be concerned primarily with the state of the art and current research directions of systems that synthesize speech from text.

Speech recognition involves people talking to machines. Speech recognizers range in sophistication from systems that recognize an isolated word or phrase to fully conversational recognizers that attempt to deal with vocabularies and syntax comparable to natural language. Also included in this broad area is *speaker recognition*, where the job of the machine is to verify the talker's claimed identity or identify the talker as someone from a fixed, known population.

Speech Coding

In the emerging digital communications environment, the transmission of digital speech at low bit rates without compromising voice quality is becoming increasingly important. Low bit rate voice will play a key role, providing new capabilities in future communications systems—e.g., voice mail sent over telephone networks, integrated voice and data transmitted over packet networks, narrow-band cellular radio, and voice encryption.

The speech coding technology to achieve high voice quality is well developed for bit rates as low as 16 kb/s.[1] Today, the major research activity is focused at bringing the rate to 4.8 kb/s and lower without degrading speech quality. The lower bit rates offer the possibility of providing end-to-end digital voice communications over dialed-up, public telephone lines.

Until recently, the real-time implementation of low-bit-rate-voice coders was a difficult and costly task. Recent advances in device technology and the availability of fast, programmable, digital signal processors (DSPs)[2] have made the task much easier. We are now able to put complex, speech processing algorithms on a single chip.[3]

Current Technology. The objective in speech coding is to transform the analog speech signal into a digital representation.

Redundancies, introduced in the speech signal during the human speech production process, make it possible to encode speech at low bit rates. Moreover, our hearing system is not equally sensitive to distortions at different frequencies and has a limited dynamic range. Speech coding techniques take advantage of these properties to reduce the bit rate.

What are our current capabilities for synthesizing high-quality speech at low bit rates?

In general, a speech coder's performance goes down with decreasing bit rates. Figure 1 illustrates the tradeoff between speech quality and bit rate for several types of speech coders. In the diagram, the speech quality is expressed as poor, fair, good, and excellent.

Speech coders can be broadly classified into two groups: waveform coders and vocoders (voice coders).

Waveform coders aim at reproducing the speech waveform as faithfully as possible. They provide high-quality synthetic speech above 16 kb/s, but their perform-

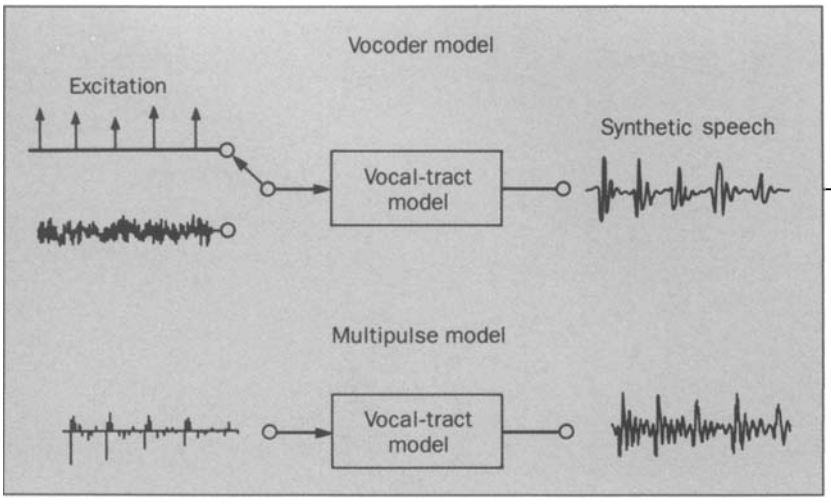

Figure 2. Traditional vocoder and multipulse models for speech synthesis.

ance usually falls off rapidly at much lower bit rates.

Vocoders use a model of human speech production to obtain a compact representation of the speech signal. Thus, they can bring the bit rate down to much lower values—even as low as 400 b/s—but the speech quality, at best, is only fair.

Today, our ability to provide high-quality speech below 8 kb/s is limited but the next generation of coders promise to fill this gap in performance. They will take advantage of the new capabilities offered by very-large-scale-integration (VLSI) technology.

Speech coding methods have been standardized both at 64 and 32 kb/s, and coders at these rates are being used in telephone networks. There are no standards yet for lower bit rates.

The 16-kb/s bit rate is suitable for a variety of applications—such as voice mail, secure voice over wideband cellular radio channels, and integrated transmission of voice and data over packet networks.

Today, the speech coding technology is available to achieve high speech quality at 16 kb/s. These coding techniques are more complex than those used in standard PCM (pulse-code modulation) and ADPCM (adaptive differential PCM) coders, but can be implemented on a single, digital signal processor chip to do real-time coding.

For example, adaptive predictive coders,[4] subband coders with adaptive bit allocation,[5] and multipulse linear predictive coders (MPLPCs)[6] that have been implemented on a single DSP chip can provide high-quality speech at 16 kb/s. Subjective tests provide a mean opinion score (MOS) of 3.9 for the multipulse coder and 3.8 for the adaptive bit allocation subband coder, both operating at 16 kb/s.

Another hybrid coder that combines the adaptive predictive and multipulse coders has produced speech at 16 kb/s with speech quality exceeding that of the ADPCM coder at 32 kb/s. A multipulse coder is capable of providing high-quality speech at rates even lower than 16 kb/s. We will describe this coder in greater detail later.

Speech Synthesis Models. To achieve high voice quality at low bit rates, we must have a speech synthesis model that can reproduce many different voices, yet requires a small amount of control information.

A synthesis model that has been popular over many years is the traditional vocoder model that generates synthetic speech by exciting a linear filter with pitch pulses or white noise. This simple vocoder model's limitations are now well known.

To overcome such limitations, the multipulse LPC model[7] replaces the traditional pitch-pulse and white-noise excitation with a sequence of pulses. Their amplitudes and locations are chosen to minimize the perceptual difference between original and synthetic speech signals. Figure 2 illustrates both the traditional vocoder and the multipulse excitation models.

The multipulse model has enough flexibility to reproduce a wide variety of speech waveforms, including voiced and unvoiced speech. The model is reasonably efficient; only a few pulses (typically 8 to 16 every 10 ms) are needed in the multipulse excitation to produce high-quality synthetic speech. A linear filter with a pitch loop can be incorporated in the synthesizer[6] to reduce the number of pulses, in particular for high-pitched voices.

Recently, another model that is based on stochastic excitation[8] has shown great promise for producing high-

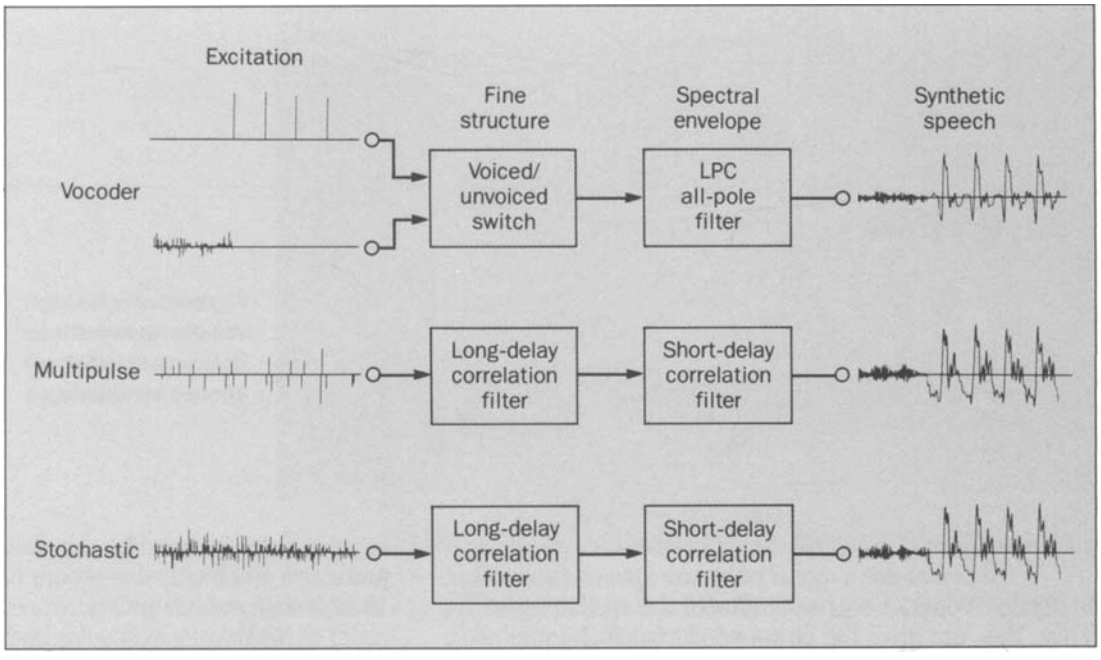

Figure 3. Different speech synthesis models.

quality speech at low bit rates. Here, the excitation is selected from a codebook of random, white Gaussian sequences using a fidelity criterion that minimizes the perceptual difference between the original and synthetic speech signals.

Figure 3 illustrates the different synthesis models. The multipulse and stochastic models use identical linear filters to introduce correlations at short and long delays in the output speech signal, but specify the excitation to the linear filter differently.

Excitation Models. Figure 4 illustrates the principle for determining the excitation in multipulse coders.

The synthetic speech signal at the synthesis filter's output is compared to the original speech signal, and the error signal is further processed to produce a measure of perceptual error. This processing includes linear filtering of the objective error to attenuate frequencies where the error is perceptually less important and amplify frequencies where the error is perceptually more important. The excitation is chosen to minimize the perceptual error.

The locations and amplitudes of pulses in the multipulse excitation are obtained sequentially, one pulse at a time. After the first pulse has been determined, a new error is computed by subtracting the first pulse's contribution to the error, and the next pulse's location is determined by finding the minimum of the new error. The process of locating new pulses continues until the error is reduced to acceptable values or the number of pulses reaches the maximum value that can be encoded at the specified bit rate.

The number of pulses determines the speech quality and the bit rate for the multipulse excitation. Four to eight pulses in a 5-ms frame are enough for producing high-quality speech.

Speech Quality Below 8 kb/s. Recent speech coding work[8] that uses stochastically excited, linear predictive coding shows great promise for producing high-quality speech below 8 kb/s and possibly as low as 4.8 kb/s. Such low rates are suitable for transmitting digital speech over narrow-band radio channels and providing end-to-end digital speech communications over ordinary dial-up, public telephone lines.

To determine the excitation in stochastic coders, an exhaustive search (Figure 5) is done from a codebook of white Gaussian sequences to minimize the perceptual distortion in the synthetic speech. These coders are extremely complex and require more than 20 million multiply/add operations per second. But the rapid progress in custom, VLSI circuits will enable us to handle this complexity in the next few years.

The stochastic coder's architecture is well suited for VLSI implementation because the search procedure

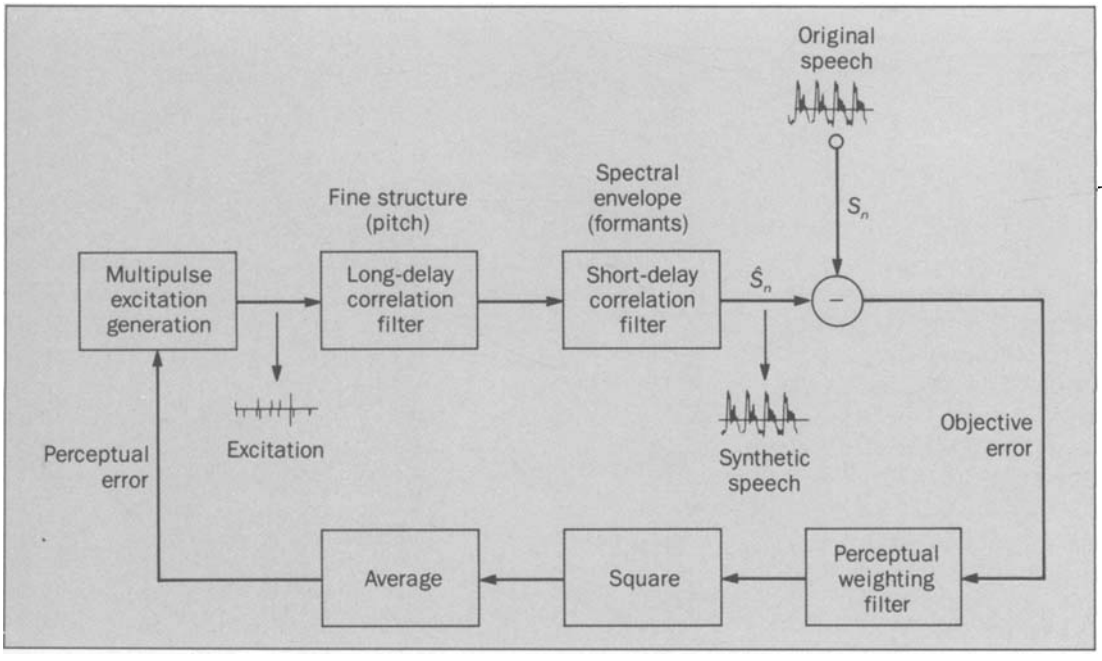

Figure 4. Procedure for determining the optimum excitation in multipulse coders.

carries out many simple, identical operations—namely, it computes error for each member of the codebook.

Stochastic coders have the potential to bring the bit rate for digital speech down to 4.8 kb/s.

Speech Synthesis

In speech synthesis, our objective is to provide a broad speaking capability to computers. Voice output has important advantages in communicating with computers. It can provide easy, remote access to information from an ordinary telephone and, when listening to speech, a person's hands and eyes remain free for other activities.

Synthetic speech can be generated in many different ways. But there are three important issues to consider here: speech quality, speaking vocabulary, and cost.

The simplest way to provide synthetic speech is to use prerecorded speech and then play it back using a speech synthesizer. Speech coding techniques can reduce the memory required for storing prerecorded speech. This method of producing voice output is economical for small- to medium-size vocabulary applications.

In other applications, such as reading electronic mail, one often needs capability for converting written text to speech. Such text-to-speech synthesis systems are much more complex than speech playback systems. Because a human talker generates prerecorded speech, these playback systems can easily produce high-quality synthetic speech. Currently, text-to-speech synthesis systems cannot produce speech of such high quality, although the quality is acceptable for many applications.

Speech Synthesis from Stored, Coded Speech. The easiest way to provide voice output on computers is to synthesize the messages from prerecorded words, phrases, and sentences spoken by a particular talker. But we must consider several tradeoffs here.

Often, choosing words as the basic synthesis unit seems proper because we can create many utterances from a few words. However, the process of joining words and creating sentences with the right prosody (pitch and duration) is much more difficult.

We can avoid this problem by recording sentences, but the number of sentences to be stored increases exponentially with the number of words in a sentence. To reduce the storage requirement, we can use a variety of speech coding methods.

Simple speech-coding procedures can produce high-quality speech at 32 kb/s, and speech coding techniques, such as multipulse LPC, can bring the data rate down to 10 kb/s. At this bit rate, a single 1-megabit ROM (read-only memory) chip can store about 100 seconds of speech data.

Multipulse LPC can produce high-quality speech

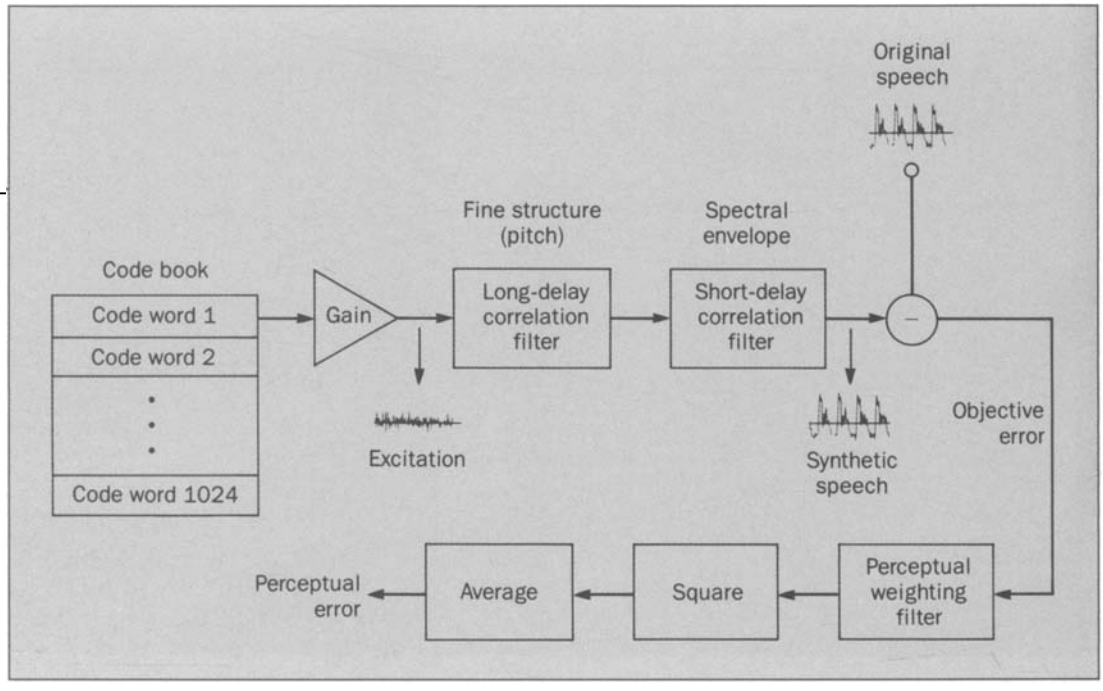

Figure 5. Search procedure for determining the best stochastic code.

using a simple speech synthesizer. Most of the complexity of multipulse LPC is in the speech analysis part that must be done only once and does not need real-time operation. With LPC vocoding techniques, we can realize data rates as low as 1 kb/s although the speech quality is much lower. (The speech is intelligible but is not natural.)

To further enhance the flexibility of stored-speech synthesis systems, we can allow control of prosody (pitch and duration adjustments) during the synthesis process. The multipulse LPC technique is particularly suitable for providing the desired control of pitch and duration. With the decreasing cost of digital storage, stored-speech synthesis techniques could provide low-cost voice output for many applications.

Text-to-Speech Synthesis. Stored-speech systems are not flexible enough to convert unrestricted English text to speech. A text-to-speech system that uses synthesis-by-rule is needed for applications such as accessing electronic mail by voice, a reading machine for the blind, and speaking proper names.

Figure 6 illustrates the various functions done by a text-to-speech system. The text-to-speech system must convert incoming text—such as electronic mail—that often includes abbreviations, Roman numerals, dates, times, formulas, and a wide variety of punctuation marks into some reasonable, standard form. The text must be further translated into a broad phonetic transcription.

To do this, we use a large pronouncing dictionary supplemented by appropriate letter-to-sound rules. The acoustic segments in a synthesis-by-rule system are short. They represent individual phonemes or transitions between neighboring phonemes that allow us to synthesize virtually unrestricted sequences of phonemes. Finally, speech waveforms are generated from acoustic parameters using multipulse LPC synthesis.

The resulting speech is intelligible and acceptable for a variety of applications.

Currently, research in text-to-speech synthesis focuses on creating speech that sounds more natural. We expect that future systems will provide more flexibility for selecting the speaker characteristics, different languages and their dialects, and regional variabilities.

Rapidly advancing VLSI technology will drastically change future speech synthesis technology. Our current computer models of speech synthesis are simple compared to what humans do, and more sophisticated synthesis models are still not practical. But future advances in technology will open new possibilities for studying a wide variety of speech data and learning to control more realistic models of human speech production.

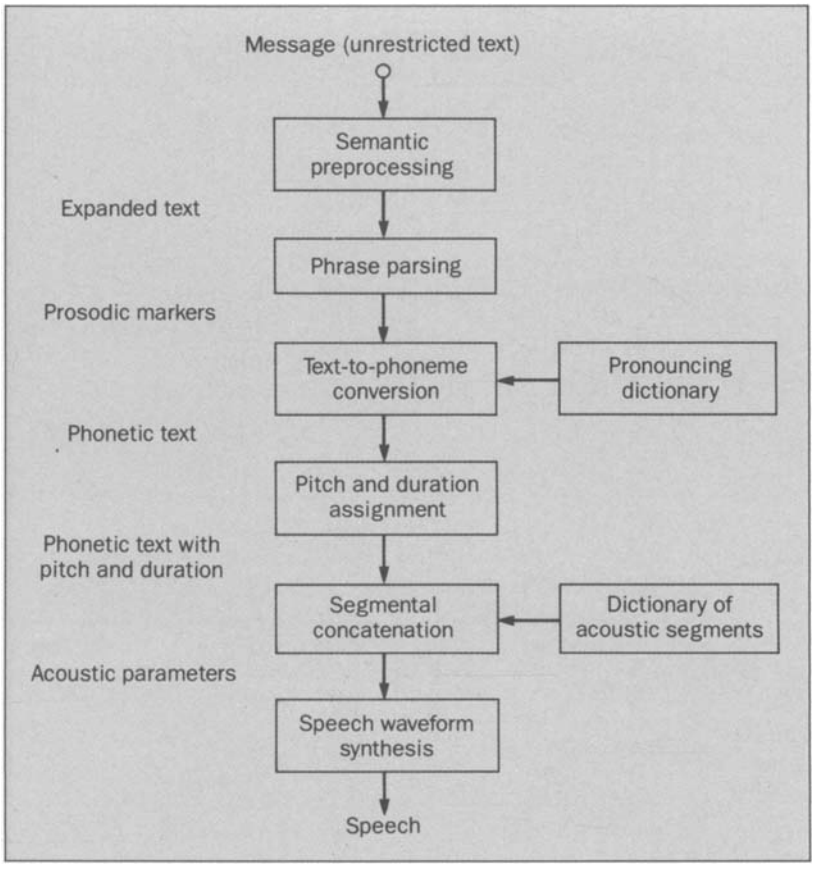

Figure 6. Various functions in a text-to-speech synthesizer.

Speech Recognition

Figure 7 shows a block diagram of the traditional, speech recognition model, which is based on pattern recognition.

The input speech signal can be anything from a single word (or a sequence of isolated words) to a sentence of continuous speech. In the first processing block, feature measurement, the speech signal is spectrally analyzed periodically to give a series of spectral feature vectors that characterize the signal's behavior. For the most part, we have used linear predictive coding as the spectral representation but other spectral analyses,[9] such as filter bank analysis, are equally suitable. The time sequence of spectral features is called a test pattern.

The second processing step is a pattern similarity measurement. Here, the running set of spectral vectors (the test pattern) is compared to a set of stored reference patterns, and a distance or similarity score is produced for each comparison. For the most part, we have used single words as the stored reference patterns. However, even when the basic recognition units are smaller than words (e.g., syllables, demisyllables, dyads, phonemes), a lexicon can be used to build word reference patterns, so we are equivalently using words as the recognition unit.

The pattern similarity measurement typically involves time registration of the stored reference pattern (a series of feature vectors) with the running speech (also a series of feature vectors). We generally use a technique known as dynamic time warping (DTW)[10] to provide the optimal alignment between references and test (speech) patterns.

Figure 8 illustrates the basic procedures of time alignment and shows representative contours of a test and reference pattern.

As the diagrams show, distinctive events in the two patterns (i.e., peaks in the contours) do not occur at the same relative instant. To derive an optimal time alignment between test and reference patterns, the DTW alignment procedure locally shrinks or expands the time axis of one pattern until it optimally matches the other pattern.

An efficient mathematical procedure[10] exists for obtaining an optimal alignment curve based on dynamic programming techniques. For the examples in Figure 8, the alignment curve appears at the upper right of the diagram. We define the similarity (or, equivalently, distance) between a reference and a test pattern as the normalized sum of the spectral similarities (distances) along the discrete set of points in the optimal time-alignment path between reference and test patterns.

The third processing step (Figure 7) is syntax control, which uses task syntax to determine the proper sequencing of stored reference patterns (words) for the task at hand. In theory, syntax control could also control the feature measurement algorithm and thus change the type (and form) of analysis depending on the sound to be recognized. However, current speech recognizers do not

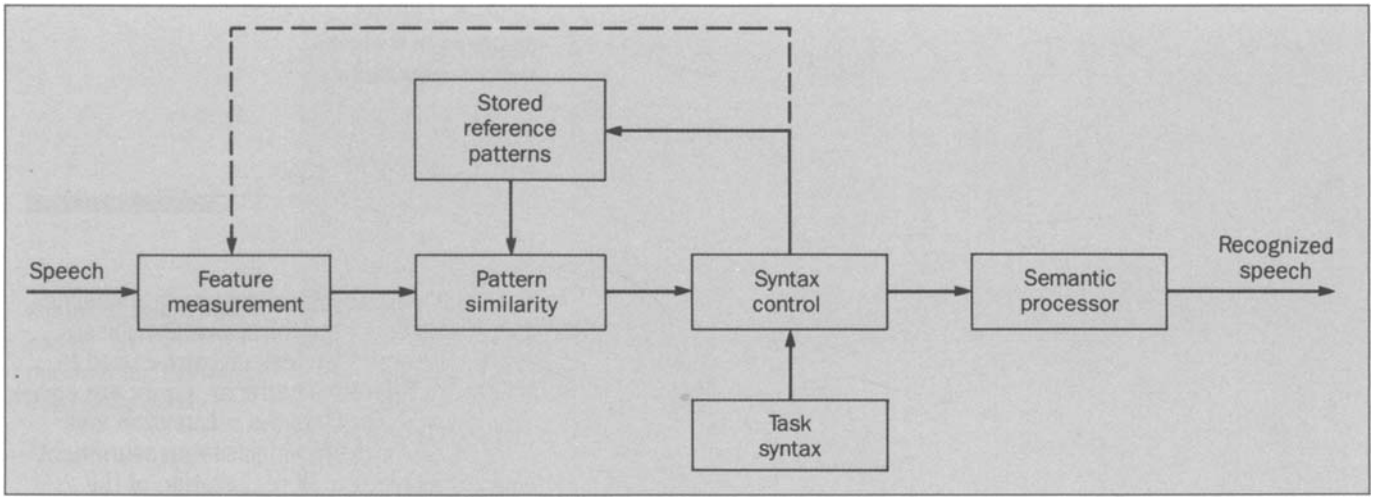

Figure 7. A speech recognizer that incorporates syntax and semantics.

use such sophisticated control.

The last processing step in Figure 7 is a semantic processor. As the recognized speech, it chooses the sentence (or word) that has the smallest distance (or highest similarity) to the input speech and is semantically meaningful. (The sentence or word has already been checked for syntax.)

HMM Models. An alternative to using templates to characterize words (or subword units) is to build probabilistic models that statistically describe the time-varying spectral characteristics of the word.

One popular form of these models is the hidden Markov model (HMM).[11] This model has N states (five appear in the example, Figure 9), and each state physically corresponds (in some vague sense) to a set of temporal events in the speech sound. The overall HMM is characterized by a state transition matrix A (that describes how new states may be reached from old states) and a statistical characterization of the acoustic vectors B (the analysis feature vectors, x) within the state.

Only minimal changes are required to the recognition structure in Figure 7 when we use HMMs rather than templates. For example, a store of reference models replaces the stored template reference patterns. Also, the pattern similarity algorithm uses statistical scoring, instead of distances, and a somewhat different alignment procedure to line up states of the reference model to frames of the test pattern.

Results with isolated words. For isolated word recognition, the classic technique has been to build both templates and statistical models that are based on natural, spoken occurrences of the word.

Table I. Performance of Isolated Word Recognizers as a Function of Vocabulary Size

Vocabulary	Speaker mode	Accuracy
10 digits	SI	99.2%
39 alphadigits	SD	79.5%
	SI	79%
54 computer terms	SI	96.5%
129 airline terms	SD	88%
	SI	91%
1109 words of basic English	SD	79.2%

NOTE: SD = speaker-dependent mode; SI = speaker-independent mode.

In the simplest case, we create a word reference pattern directly from one or more occurrences of the word spoken by a given talker. In a more sophisticated application, we cluster a set of multiple occurrences of the word to give one (or more) word reference patterns. The patterns may be talker specific, the so-called speaker dependent (SD) recognizers, or speaker independent (SI), depending on the way they are derived.

Isolated word systems have been tested with vocabularies that range from a few words (e.g., ten digits) up to over 1000 words (e.g., 1109 words of basic English).

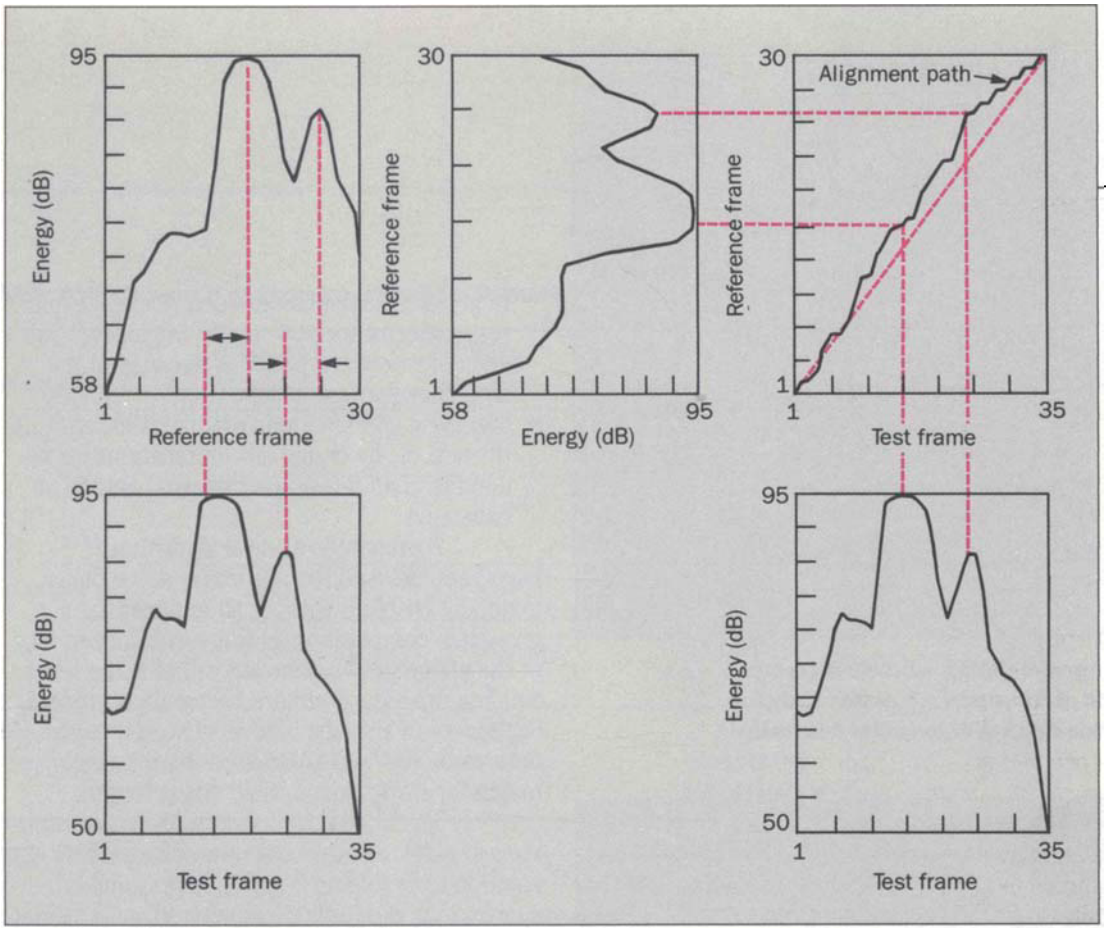

Figure 8. Time alignment between a test and reference pattern. The alignment path for the representative contours of the test and reference patterns is at the upper right.

Table I summarizes current SD and SI performance for a range of vocabularies. One can readily see that the complexity of the words in the vocabulary (i.e., how similar are the nearest sounding word pairs) is more important than mere vocabulary size.

For the 129-word airline vocabulary, the accuracy was 88 percent in a speaker-dependent mode and 91 percent in a speaker-independent mode. The improvement in accuracy of the speaker-independent system is a result of the variability of many of the words in the vocabulary. In a speaker-dependent mode, if a word is pronounced differently from the way it was spoken when the recognizer was trained, an error is almost sure to occur. Because in a speaker-independent mode there are 12 reference patterns, it is unlikely that a given word pronunciation will not be represented in one or more of the 12 patterns.

Connected Word Recognition. A somewhat more complicated task in speech recognition is to recognize speech that is nominally spoken as a connected word string, e.g., digit strings for dialing telephone numbers or letter strings for spelling names.

Figure 10 shows how we recognize such strings, using the statistical pattern recognition approach. We assume that each word in the vocabulary is represented by one or more reference patterns (i.e., templates or statistical models). Further, we can recognize the unknown spoken word string by finding the best concatenation of reference patterns that matches the test pattern. There are several problems associated with finding the optimal matching sequence of reference patterns, including:

- Generally, the number of words in the test pattern is unknown.
- The locations, in time, of the boundaries between words are unknown. Often, there really are no well-defined boundaries because the end of one word may merge smoothly with the beginning of the next word.

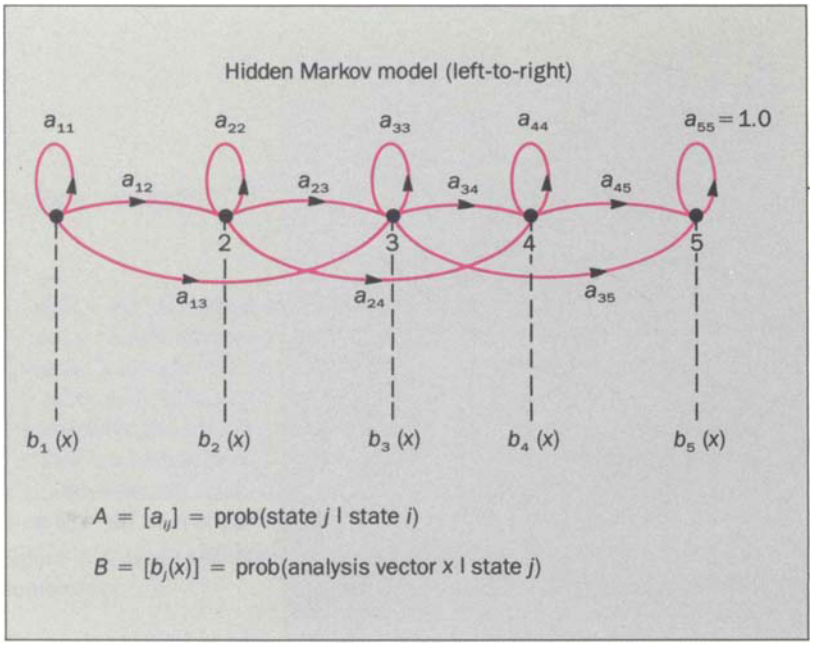

Figure 9. A hidden Markov model (HMM) suitable for representing a single word. Each of the model's N states (only five are shown here) corresponds to a set of temporal events in the speech sound.

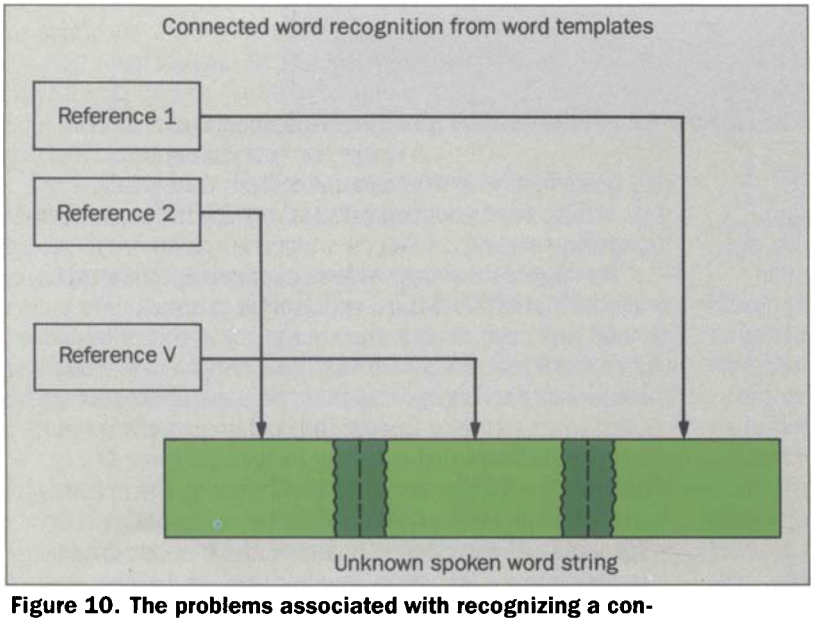

Figure 10. The problems associated with recognizing a connected word string from single word reference patterns. The end of a word often merges smoothly with the start of the next word.

- In general, matches between reference and test patterns are poor at the beginnings and ends of reference patterns because of the high degree of variability.
- Matching combinations of strings exhaustively (i.e., by trying all combinations for all lengths of all reference patterns) is too expensive.

Fortunately, several algorithms[12-16] have been devised that optimally solve the matching problem without an exponential growth in computation as the vocabulary or size of the string grows. One algorithm is the level building procedure, where recognition processing occurs in a series of levels (words) to determine the best connected-word string match for every permissible string length.

Thus, we have found solutions to problems 1 and 4, above, but know of no perfect solution to problems 2 and 3. A reasonable approach, and one that has worked well to date, is to extract word reference patterns from tokens obtained from connected word strings. For example, we can get reference patterns for digits from analysis of a training set of connected digit strings. Each reference pattern has information about the spectral dynamics of digits in strings, rather than in isolation.

Results with Connected Words. Table II summarizes current performance of connected word recognizers based on the level building algorithm.

With a digits vocabulary in a speaker-trained mode, we have obtained string accuracies greater than 98 percent for unknown and variable length strings of one to seven digits. In a speaker-independent mode, the best string accuracy has been only about 90 percent for unknown length strings.

The table also presents results for

Table II. Performance of Connected Word Recognizers

Vocabulary	Speaker mode	Word accuracy	Task	String (task) accuracy
10 digits	SD	>99%	Random strings, 1 to 7 digits	>98% UL >99% KL
	SI	97.5%		>90% UL >95% KL
26 letters	SD	~80%	Directory listing retrieval of 17,000 names	96%
	SI	~80%		90%
127 airline terms	SD	96%	Airlines reservation and information	87%
	SI	93%		75%

NOTE: UL = strings of unknown length; KL = strings of known length; SD = speaker-dependent mode; SI = speaker-independent mode.

connected letter recognition for both spelled names from a 17,000 name directory, and an airlines reservation and information task with a 127-word vocabulary.

Continuous Speech Recognition. Based on experience with more limited, speech recognition tasks, work has started on a continuous-speech recognition system, with a large vocabulary (1000 to 20,000 words) and natural syntax (i.e., approaching spoken English). Figure 11 is a block diagram of the proposed system architecture.

The following are major complications when building such a recognizer:

- Words cannot be the basic unit for recognition. Instead, we must use subword units such as syllables, demisyllables, diphones, dyads, and phonemes.
- It must use a lexicon that describes how words are made up from the subword units. The lexicon can be an explicit representation (e.g., a dictionary of pronunciations from subword units), or it can be probabilistic in nature (e.g., a statistical model).
- A language model describes the constraints among words in the language. This model could be a formal grammar, a statistical model, or even an explicit state diagram of task syntax as used in Figure 7.

Each complication listed is formidable and leads to a wide range of choices of how to handle the problem. Taken together, they suggest why continuous speech recognition is, and will remain, an unsolved problem for a long time.

Speaker Recognition

The speaker recognition problem is really a pair of problems:

- *Speaker identification*—A talker is identified as one of a given set of talkers.
- *Speaker verification*—The talker gives both a claimed identity and a transaction request, and the system decides whether to accept or reject the identity claim.

Clearly, speaker identification is a much harder problem than speaker verification. As the number of speakers increases without bound, the probability of iden-

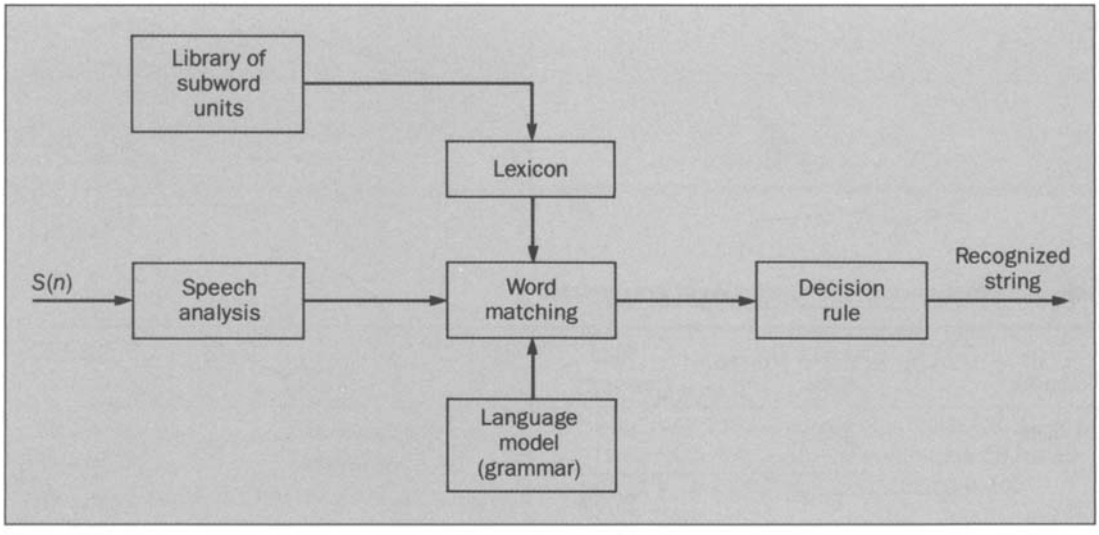

Figure 11. Model for large-vocabulary speech recognition based on subword speech units.

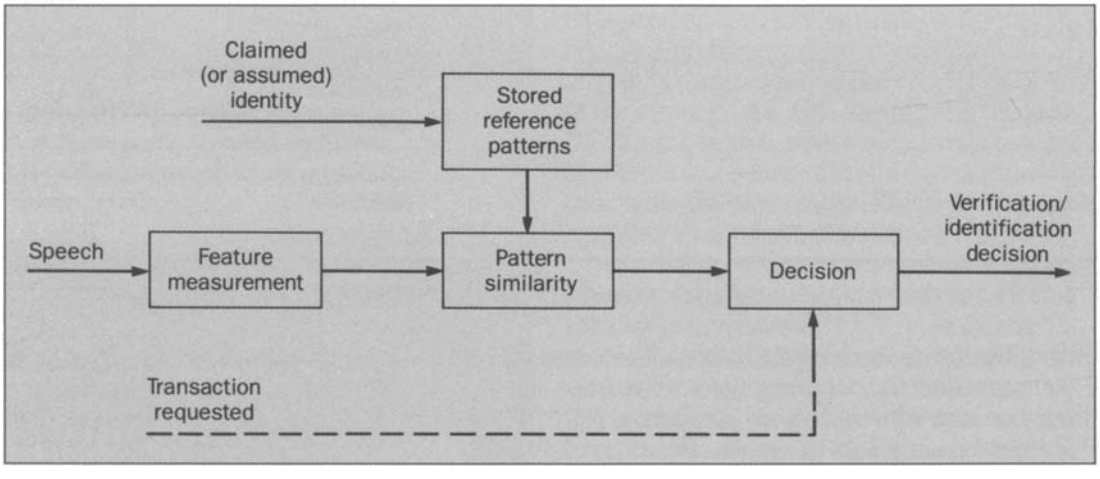

Figure 12. Block diagram of a speaker recognition system. Decision processing accepts or rejects the claimed identity, based on the results of pattern similarity processing and the transaction requested.

tifying a talker incorrectly goes to one (100-percent error), whereas the probability of error remains constant for speaker verification.

Figure 12 shows a block diagram of a speaker recognition system. The input speech, which can be either a sentence or a sequence of words (e.g., digits), is first spectrally analyzed. Then, the resulting spectral pattern is compared to stored reference patterns, using DTW methods.

For speaker identification, the pattern similarity processing must be done for each assumed talker (i.e., for the entire set of talkers), and the *decision* box chooses the identified talker as the one with the highest similarity to the input speech.

For speaker verification, the pattern similarity processing is only done for the claimed identity; i.e., there is only one distance score. Based on the transaction requested and the similarity score of the DTW processing, the decision box decides whether to accept or reject the claimed identity. Thus, a banking transaction would require a lower degree of similarity for the speaker to deposit money into an account than to withdraw money from it.

By comparing Figures 7 and 12, we can see that the processing for speech and speaker recognition is similar. Thus, as fundamental improvements are made in any basic procedure (feature measurement, pattern similarity, etc.), the performance of both types of systems improves.

Key factors that affect the performance of speaker verification systems are the type of input string used, the features used to characterize the voice pattern, and the

Table III. Performance of Speaker Verification Systems

Input	Mode	Acoustic signal processing	Feature pattern	Verification performance (equal-error rate)
Sentence-long utterances	Text dependent	Tenth-order cepstral analysis	Time contours of cepstral coefficients	1% recorded-telephone
				4% live telephone
Isolated word strings	Text independent	Eighth-order LPC analysis	Speaker-dependent word templates	4% recorded-telephone
			Speaker-independent template distance	8% recorded-telephone
Isolated word strings	Text independent	Eighth-order cepstral analysis	Vector quantization codebook talker models	1% recorded-telephone

type of transmission system over which the verification system is used.

The best performance is achieved when sentence-long utterances are used in a noise-free speaking environment. Conversely, poorer performance is achieved for short, unconstrained utterances, spoken in a noisy environment. Table III summarizes current performance of several types of speaker verification systems.[17-18] Current research in speaker recognition aims to improve performance by periodically adapting the talker patterns to track changes in voice patterns.

References

1. N. S. Jayant, "Coding speech at low bit rates," *IEEE Spectrum*, Vol. 23, No. 8, pp. 58-63, August 1986.
2. W. P. Hayes et al., "A 32-bit VLSI digital signal processor," *IEEE Journal of Solid-State Circuits*, Vol. SC-20, No. 5, October 1985, pp. 998-1004.
3. H. Alrutz, "Implementation of a multi-pulse coder on a single chip floating-point signal processor," *Proceedings of the 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, April 1986, pp. 2367-2370.
4. B. S. Atal, "Predictive coding of speech at low bit rates," *IEEE Transactions on Communications*, Vol. COM-30, April 1986, pp. 600-614.
5. T. A. Ramstad, "Sub-band coder with a simple adaptive bit allocation algorithm," *Proceedings of the 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paris, France, April 1982, pp. 203-207.
6. S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC coders at low bit rates," *Proceedings of the 1984 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, paper No. 1.3, March 1984.
7. B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proceedings of the 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paris, France, 1982, pp. 614-617.
8. B. S. Atal and M. R. Schroeder, "Stochastic coding of speech signals at very low bit rates," *Proceedings of the International Conference on Communications—ICC84*, Part 2, May 1984, pp. 1610-1613.
9. J. D. Markel and A. H. Gray Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.
10. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, February 1975, pp. 66-72.

11. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *The Bell System Technical Journal*, Vol. 62, No. 4, April 1983, pp. 1035-1074.
12. H. Sakoe, "Two Level DP Matching—A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, December 1979, pp. 588-595.
13. C. S. Myers and L. R. Rabiner, "Connected Digit Recognition Using a Level Building DTW Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 3, June 1981, pp. 351-363.
14. J. S. Bridle, M. D. Brown, and R. M. Chamberlain, "An Algorithm for Connected Word Recognition," *Automatic Speech Analysis and Recognition*, J. P. Haton, Ed., D. Reidel Publishing Company, Dordrecht, Holland, 1982, pp. 191-204.
15. J. L. Gauvain and J. Mariani, "A Method for Connected Word Recognition and Word Spotting on a Microprocessor," *Proceedings of the 1982 ICASSP*, May 1982, pp. 891-894.
16. L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A Segmental *k*-Means Training Procedure for Connected Word Recognition," *AT&T Technical Journal*, Vol. 65, No. 3, May/June 1986, pp. 21-31.
17. S. Furui, "Cepstrum Analysis Technique for Automatic Speaker Verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 2, April 1981, pp. 254-272.
18. F. K. Soong, A. E. Rosenberg, L. R. Rabiner, and B. H. Juang, "A Vector Quantization Approach to Speaker Recognition," *Proceedings of the ICASSP '85*, April 1985, pp. 387-390.

(Manuscript received August 4, 1986)