Interactively Skimming Recorded Speech

Barry Michael Arons

B.S. Civil Engineering, Massachusetts Institute of Technology, 1980
M.S. Visual Studies, Massachusetts Institute of Technology, 1984

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at the
Massachusetts Institute of Technology

February 1994

Author:
____
Program in Media Arts and Sciences
January 7, 1994

Certified by:
____
Christopher M. Schmandt
Principal Research Scientist
Thesis Supervisor

Accepted by:
____
Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Interactively Skimming Recorded Speech

Barry Michael Arons

Submitted to the Program in Media Arts and Sciences on January 7, 1994, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Listening to a speech recording is much more difficult than visually scanning a document because of the transient and temporal nature of audio. Audio recordings capture the richness of speech, yet it is difficult to directly browse the stored information. This dissertation investigates techniques for structuring, filtering, and presenting recorded speech, allowing a user to navigate and interactively find information in the audio domain. This research makes it easier and more efficient to listen to recorded speech by using the *SpeechSkimmer* system.

First, this dissertation describes *Hyperspeech,* a speech-only hypermedia system that explores issues of speech user interfaces, browsing, and the use of speech as data in an environment without a visual display. The system uses speech recognition input and synthetic speech feedback to aid in navigating through a database of digitally recorded speech. This system illustrates that managing and moving in time are crucial in speech interfaces. Hyperspeech uses manually segmented and structured speech recordings—a technique that is practical only in limited domains.

Second, to overcome the limitations of Hyperspeech while retaining browsing capabilities, a variety of speech analysis and user interface techniques are explored. This research exploits properties of spontaneous speech to automatically select and present salient audio segments in a time-efficient manner. Two speech processing technologies, *time compression* and *adaptive speech detection* (to find hesitations and pauses), are reviewed in detail with a focus on techniques applicable to extracting and displaying speech information.

Finally, this dissertation describes *SpeechSkimmer,* a user interface for interactively skimming speech recordings. SpeechSkimmer uses simple speech processing techniques to allow a user to hear recorded sounds quickly, and at several levels of detail. User interaction, through a manual input device, provides continuous real-time control of the speed and detail level of the audio presentation. SpeechSkimmer incorporates time-compressed speech, pause removal, automatic *emphasis detection*, and non-speech audio feedback to reduce the time needed to listen. This dissertation presents a multi-level structural approach to auditory skimming, and user interface techniques for interacting with recorded speech.

Thesis Supervisor: Christopher M. Schmandt
Principal Research Scientist, Program in Media Arts and Sciences

This work was performed in the Media Laboratory at MIT. Support for this research was provided, in part, by Apple Computer, Inc., Interval Research Corporation, and Sun Microsystems, Inc. The ideas expressed herein do not necessarily reflect those of the supporting agencies.

Doctoral Dissertation Committee

Thesis Advisor: ____

Christopher M. Schmandt
Principal Research Scientist
Program in Media Arts and Sciences

Thesis Reader: ____

Nathaniel I. Durlach
Senior Research Scientist
Department of Electrical Engineering and Computer Science

Thesis Reader: ____

Thomas W. Malone
Patrick J. McGovern Professor of Information Systems
Sloan School of Management

Acknowledgments

Chris Schmandt and I have been building interactive speech systems together for over a decade. Chris is one of the founding fathers of the conversational computing field, an area of emerging importance that has kept me motivated and busy over the years. Thanks for helping me get here.

Nat Durlach has been a friendly and supportive influence, challenging me to think about how people operate and how we can get machines to communicate with them efficiently and seamlessly.

Tom Malone has provided much encouragement and insight into the way humans, machines, and interfaces should work in the future.

Don Jackson, from Sun Microsystems, provided support and friendship since before this dissertation process even started.

Eric Hulteen, from Apple Computer, critiqued many of my early ideas and helped shape their final form.

David Liddle and Andrew Singer, from Interval Research, got me involved in what could be the best place since the ArcMac.

Lisa Stifelman provided valuable input in user interface design, assisted in the design and the many hours of carrying out the usability test, taught me the inner workings of the Macintosh, and helped edit this document.

Meg Withgott lent her help and expertise on pitch, emphasis, and segmentation.

Michele Covell and Michael Halle provided technical support (when I needed it most) beyond the call of duty. I hope to repay the favor someday.

Special thanks to Linda Peterson for her time, patience, and help in keeping me (and the entire Media Arts and Sciences program) on track.

Steve Benton provided behind the scenes backing and a helping hand (or signature) when it was needed.

Gayle Sherman provided advice, signatures, and assistance in managing the bureaucracy at the Media Laboratory and MIT.

Doug Reynolds ran his speaker identification software on my recordings. Don Hejna provided the SOLAFS implementation. Jeff Herman integrated the BBC radio data into SpeechSkimmer.

Thanks to Marc Davis, Abbe Don, Judith Donath, Nicholas Negroponte, and William Stasior for their assistance, advice, and for letting me use recordings of their speech. Tom Malone's spring 1993 15.579A class also allowed me to record parts of two of their discussions.

Jonathan Cohen and Gitta Salomon provided important help along the way. Walter Bender, Janet Cahn, George Furnas, Andrew Kass, Paul Resnick, Louis Weitzman, and Yin Yin Wong also contributed to this work in important ways.

Portions of this dissertation originally appeared in ACM conference proceedings as Arons 1991a and Arons 1993a, Copyright 1991 and 1993, Association of Computing Machinery, Inc., reprinted by permission. Parts of this work originally appeared in AVIOS conference proceedings as Arons 1992a and Arons 1991b, reprinted by permission of the American Voice I/O Society.

When I completed my Master of Science degree in 1984 the idea of going on to get a doctorate was completely out of the question, so I dedicated my thesis to the memory of my parents. I didn't need or want a Ph.D., writing was a struggle, and I wanted to get out of MIT. It is now ten years later, I'm back from the west coast, and I *am* completing my dissertation. Had I known, I would have waited and instead dedicated this dissertation to them for instilling me with a sense of curiosity, a passion for learning and doing, and the insight that has brought me to this point in my life.

Contents

Figures

1 Motivation and Related Work

As life gets more complex, people are likely to read less and listen more.
(Birkerts 1993, 111)

Speech has evolved as an efficient means of human-to-human communication, with our vocal output reasonably tuned to our listening and cognitive capabilities. While we have traditionally been constrained to listen only as fast as someone speaks, the ancient Greek philosopher Zeno said “Nature has given man one tongue, but two ears so that we may hear twice as much as we speak.” In the last 40 years technology has allowed us to increase our speech listening rate, but only by a factor of about two. This dissertation postulates that through appropriate processing and interaction techniques it is possible to overcome the time bottleneck traditionally associated with using speech—that we can skim and listen many times faster than we can speak.

This research addresses issues of accessing and listening to speech in new ways. It reviews a variety of areas related to high-speed listening, presents two technical explorations of skimming and navigating in speech recordings, and provides a framework for thinking about such systems. This work is not an incremental change from what exists today—the techniques and user interfaces presented herein are a whole new way to think about speech and audio.

Important ideas addressed by this work include:

- The importance of time when listening to and navigating in recorded speech.
- Techniques to provide multi-level structural representations of the content of speech recordings.
- User interfaces for accessing speech recordings.

This chapter describes why skimming speech is an important but difficult issue, reviews background material, and provides an overview of this research.

1.1 Defining the Title

The most difficult problem in performing a study on speech-message information retrieval is defining the task.

(Rose 1991, 46)

The title of this work, *Interactively Skimming Recorded Speech,* is important for a variety of reasons. It has explicitly been made concise to describe and exemplify what this document is about—all unnecessary words and redundancies were removed. Working backward through the title, each word is defined in the context of this document:

Speech is "the communication or expression of thoughts in spoken words" (Webster 1971, 840). Although portions of this work are applicable to general audio, "speech" is used because it is our primary form of interpersonal communication, and the research focuses on exploiting information that exists only in the speech channel.

Recorded indicates that speech is stored for later retrieval. The form and format of the storage are not important, although a random access digital store is used and assumed throughout this document. While the systems described run in "real time," they must be used on existing speech. While someone is speaking, it is possible to review what they have already said, but it is impossible to browse forward in time beyond the current instant.

Skimming means "to remove the best … contents from" and "to pass lightly or hastily: glide or skip along, above, or near a surface" (Webster 1971, 816). Skim is used to mean quickly extracting the salient information from a speech recording, and is similar to *browsing* where one wanders around to see what turns up. Skimming and browsing techniques are often used while *searching*—where one is looking for a particular piece of information. *Scanning* is similar to skimming in many ways, yet it connotes a more careful examination using vision. Skimming here is meant to be performed with the ears. Note that the research concentrates on skimming rather than summarization, since it is most general, computationally tractable, and applicable to a wide range of problems.

Interactively indicates that the speech is not only presented, but segments of speech are selected and played through the mutual actions of the listener and the skimming system. The listener is an active and critical component of the system.

1.2 Introduction

> *Reading, because we control it, is adaptive to our needs and rhythms.… Our ear, and with it our whole imaginative apparatus, marches in lockstep to the speaker's baton.*
>
> (Birkerts 1993, 111)

Skimming, browsing, and searching are traditionally considered visual tasks. One easily performs them while reading a newspaper, window shopping, or driving a car. However, there is no natural way for humans to skim speech information because of the transient nature of audio—*the ear cannot skim in the temporal domain the way the eyes can browse in the spatial domain.*

Speech is a powerful communications medium—it is, among other things, natural, portable, and rich in information, and it can be used while doing other things. Speech is efficient for the talker, but is usually a burden on the listener (Grudin 1988). It is faster to speak than it is to write or type; however, it is slower to listen than it is to read. This research integrates information from multiple sources to overcome some of the limitations of listening to speech. This is accomplished by exploiting some of the regular properties of speech to enable high-speed skimming.

1.2.1 The Problem: Current User Scenarios

Recorded audio is currently used by many people in a variety of situations including:

- lectures and interviews on microcassette
- voice mail
- tutorial and motivational material on audio cassette
- conference proceedings on tape
- books on tape
- time-shifted, or personalized, radio and television programs

Such "personal" uses are in addition to commercial uses such as:

- story segments for radio shows
- using the audio track for editing a video tape story line
- information gathering by law enforcement agencies

This section presents some everyday situations that demonstrate the problems of using stored speech that are addressed by this research.

1.2.1.1 Searching for Audio in Video

> *When browsing the meeting record sequentially, it is convenient to replay it in meaningful units. In a medium such as videotape, this can be difficult since there is no way of identifying the start of a meaningful unit. When fast forwarding through a videotape of the meeting, people [reported they] … frequently ended up in the middle of a discussion rather than the start.*
>
> (Wolf 1992, 4)

There are important problems in the field of video production, logging, and editing that are better addressed in the audio domain than in the visual domain (Davenport 1991; Pincever 1991). For example, after a television reporter and crew conduct an interview for the nightly news they edit the material under tight time and content constraints. After the interview, the reporter's primary task is typically to find the most appropriate "sound bite" before the six o'clock news. This is often done in the back of a news van using the reporter's hand-scribbled notes, and by searching around an audio recording on a microcassette recorder.

Finding information on an audio cassette is difficult. Besides the fact that a tape only provides linear access to the recording, there are several confounding factors that make browsing audio difficult. Although speaking is faster than writing or typing, listening to speech is slow compared to reading. Moreover, the ear cannot browse an audio tape. A recording can be sped up on playback; however on most conventional tape players this is accompanied by a change in pitch, resulting in a loss of intelligibility.

Note that the task may not be any easier if the reporter has a videotape of the interview. The image of a "talking head" adds few useful cues since *the essential content information is in the audio track.* When the videotape is shuttled at many times normal speed an identifiable image can be displayed, yet the audio rapidly becomes unintelligible.

1.2.1.2 Lecture Retrieval

When attending a presentation or a lecture, one often takes notes by hand or with a notebook computer. Typically, the listener has to choose between listening for the sake of understanding the high-level ideas of the speaker or taking notes so that none of the low-level details are lost. The listener may attempt to capture one level of detail or the other, but because of time constraints it is often difficult to capture both perspectives. Few people make audio recordings of talks or lectures. It is much easier to browse one's hand-written notes than it is to listen to a recording. *Listening is a time-consuming real time process.*

If an audio recording is made, it is difficult to access a specific piece of information. For example, in a recorded lecture one may wish to review a short segment that describes a single mathematical detail. The listener has two choices in attempting to find this small chunk of speech: the entire tape can be played from the beginning until the desired segment is heard, or the listener can jump around the tape attempting to find the desired information. The listener's search is typically inefficient, time-consuming, and frustrating because of the linear nature of the tape and the medium. In listening to small snippets of speech from the tape, it is difficult to find audio landmarks that can be used to constrain the search. Users may try to minimize their search time by playing only short segments from the tape, yet they are as likely to play an unrelated comment or a pause as they are to stumble across an emphasized word or an important phrase. It is difficult to perform an efficient search even when using a recorder that has a tape counter or time display.

This audio search task is analogous to trying to find a particular scene in a video tape, where the image can only be viewed in play mode (i.e., the screen is blank while fast forwarding or rewinding). The user is caught between the slow process of looking or listening, and an inefficient method of searching in the visual or auditory domain.

1.2.2 Speech Is Important

Speech is a rich and expressive medium (Chalfonte 1991). In addition to the lexical content of our spoken words, our emotions and important syntactic and semantic information are captured by the pitch, timing, and amplitude of our speech. At times, more semantic information can be transmitted by the use of silence than by the use of words. Such information is difficult to convey in a text transcription, and is best captured in the sounds themselves.

Transcripts are useful in electronically searching for keywords or visually skimming for content. Transcriptions, however, are expensive—a one hour recording of carefully dictated business correspondence takes at least an hour to transcribe and will usually cost roughly $20 per hour. A one hour interactive meeting or interview will often take over six hours to transcribe and cost over $150. Note that automatic speech recognition-based transcriptions of spontaneous speech, meetings, or conversations are not practical in the foreseeable future (Roe 1993).

Speech is becoming increasingly important for I/O and for data storage as personal computers continue to shrink in size. Screens and keyboards lose their effectiveness in tiny computers, yet the transducers needed to capture and play speech can be made negligible in size. Negroponte said:

> The … consequence of this view of the future is that the form factor of such *dynadots* suggests that the dominant mode of computer interaction will be speech. We can speak to small things. (Negroponte 1991, 185)

1.2.3 Speech Storage

Until recently, the use of recorded speech has been constrained by storage, bandwidth, computational, and I/O limitations. These barriers are quickly being overcome by recent advances in electronics and related disciplines, so that it is now becoming technologically feasible to record, store, and randomly access large amounts of recorded speech. Personal computers and workstations are now capable of recording and playing audio, regularly contain tens or hundreds of megabytes of RAM, and can store tens of gigabytes of data on disks.

As the usage scenarios in section 1.2.1 illustrate, recorded speech is important in many existing interfaces and applications. Stored speech is becoming more important as electronics and storage costs continue to decrease, and portable and hand-held computers ("personal digital assistants" and "personal communicators") become pervasive. Manufacturers are supplying the base hardware and software technologies with these devices, but *there is currently no means for interacting with, or finding information in, large amounts of stored speech.*

One interesting commercial device is the Jbird digital recorder (Adaptive 1991). This portable pocket-sized device is a complete digital recorder, with a signal processing chip for data compression, that stores up to four hours of digitized audio to RAM. This specialized device is designed to be used covertly by law enforcement agencies, but similar devices may eventually make it to the consumer market as personal recording devices and memory aids (Lamming 1991; Stifelman 1993; Weiser 1991).

1.2.4 A Non-Visual Interface

If you were driving home in your car right now, you couldn't be reading a newspaper.
Heard during a public radio fund drive.

Speech is fundamental for human communication, yet this medium is difficult to skim, browse, and navigate because of the transient nature of audio. In displaying a summary of a movie, television show, or home video one can show a time line of key frames (Davis 1993; Mills 1992) or a video extrusion (Elliott 1993), possibly augmented with text or graphics, that provides a visual context. It is not possible to display a conversation or radio show in an analogous manner. If the highlights of the radio program were to be played simultaneously, the resulting cacophony would be neither intelligible nor informative.

A waveform, spectrogram, or other graphical representation can be displayed, yet this provides little content information.[1] Text tags (or a full transcription) can be shown; however, this requires an expensive transcoding from one medium to another, and causes the rich attributes of speech to be lost. Displaying speech information graphically for the sake of finding information in the signal is somewhat like taking aspirin for a broken arm—it makes you feel better, but it does not attack the fundamental problem.

This research therefore concentrates on *non-visual*, or *speech-only interfaces* that do not use a display or a keyboard, but take advantage of the audio channel. A graphical user interface may make some speech searching and skimming tasks easier, but there are several reasons for exploring non-visual interfaces. First, there are a variety of situations where a graphical interface cannot be used, such as while walking or driving an automobile, or if the user is visually impaired. Second, the important issue addressed in this research is structuring and extracting information from the speech signal. Once non-visual techniques are developed to extract and present speech information, they can be taken advantage of in visual interfaces. However, tools and techniques learned from graphical interfaces are less applicable to non-visual interfaces.

[1] A highly trained specialist can slowly "read" spectrograms; however, this approach is impractical and slow, and does not provide the cues that make speech a powerful communication medium.

1.2.5 Dissertation Goal

The focus of this research is to provide simple and efficient methods for skimming, browsing, navigating and finding information in speech interfaces. Several things are required to improve information access and add skimming capabilities in scenarios such as those described in section 1.2.1. Tools and algorithms, such as time compression and semantically based segmentation, are needed to enable high-speed listening to recorded speech. In addition, user interface software and technologies must be developed to allow a user to access the recorded information and control its presentation.

This dissertation addresses these issues and problems by helping users skim and navigate in speech. Computer-based tools and interaction techniques have been developed to assist in interactively skimming and finding information purely in the audio domain. This is accomplished by matching the system output to the listener's cognitive and perceptual capabilities. The focus of this research is not on developing new fundamental speech processing algorithms,[2] but to combine interaction techniques with speech processing technologies in novel and powerful new ways. The goal is to provide auditory "views" of speech recordings at different time scales and abstraction levels *under interactive user control*—from a high-level audio overview to a detailed presentation of information.

1.3 Skimming this Document

> *auditory information is temporally fleeting: once uttered, special steps have to be taken to refer to it again, unlike visually presented information that may be referred to at leisure.*
>
> (Tucker 1991, 148)

This section provides a brief road map to the dissertation research and this document, encouraging quick visual skimming in areas of interest to the reader. The number of bullets indicate the chapters that a reader should consider looking at first.

- ••• Casual readers will find chapter 1, particularly section 1.2 most interesting, as it describes the fundamental problems being addressed and the general approach to their solution through several user scenarios.

[2] However, they were developed where needed.

- Chapter 2 describes Hyperspeech, a preliminary investigation of speech-only browsing and navigation techniques in a manually authored hypermedia database. The development of this system inspired the research described in this dissertation.
- Chapter 3 reviews methods to time compress speech, including perceptual limits, and the significance and importance of pauses in understanding speech.
- Chapter 4 reviews techniques of finding speech versus background noise in recordings, focusing on techniques that can be used to segment recordings, and methods that adapt to different background noise levels.
- Chapter 5 describes SpeechSkimmer, a user interface for interactively skimming recorded speech. Section 5.9 details algorithms developed for segmenting speech recordings based on pauses and on pitch.
- Chapter 6 discusses the contributions of the research, and areas for continued work.

1.4 Related Work

This research draws from diverse disciplines, integrating theories, ideas, and techniques in important new ways. There is a wide body of knowledge and literature that addresses particular aspects of this problem; however, none of them provides an adequate solution to navigating in recorded speech.

Time compression technologies allow the playback speed of a recording to be increased, but there are perceptual limits to the maximum speed increase. The use of time-compressed speech plays a major role in this dissertation and is reviewed in detail in chapter 3.

There has been some work done in the area of *summarizing* and *gisting* (see section 1.4.2), but these techniques have been constrained to limited domains. Research on *speech interfaces* has laid the groundwork for this exploration by providing insight into the problems of using speech, but has not directly addressed the issue of finding information in speech recordings. There has been significant work in *presenting* and *retrieving information,* but this has focused on textual information and graphical interfaces.

1.4.1 Segmentation

There are several techniques for segmenting speech, but these have not been applied to the problem of skimming and navigating. Chapter 2 describes some manual and computer-assisted techniques for segmenting recorded speech. A speech detector that determines the presence or absence of speech can be effective at segmenting recordings. A variety of speech detection techniques are described in chapter 4. Sections 5.9.3 and 5.9.5 describe other segmentation techniques based on pauses and the fundamental frequency of speech.

Kato and Hosoya investigated several techniques to enable fast message searching in telephone-based information systems (Kato 1992; Kato 1993). They broke up messages on hesitation boundaries, and presented either the initial portion of each phrase or segments based on high energy portions of speech. They found that combining these techniques with time compression enabled fast message browsing.

Hawley describes many techniques and algorithms for extracting structure out of sound (Hawley 1993). Hawley's work focuses on finding music and speech in audio recordings with an eye towards parsing movie sound tracks.

Wolf and Rhyne present a method for selectively reviewing meetings based on characteristics captured by a computer-supported meeting tool (Wolf 1992). They found the temporal pattern of workstation-based turn-taking to be a useful index to points of interest within the meeting log. They did this by analyzing patterns of activity that are captured by the computerized record of the meetings. They then attempted to characterize the structure of the meetings by correlating these data with points of interest to assist in reviewing the meeting. The intent of each turn taken during a meeting was coded into one of five categories. The turn categories of most interest for assisting in browsing the meeting record were preceded by longer gaps than the other turn types. They found, for example, that finding parts following gaps of ten seconds or longer provides a more efficient way of browsing the meeting record than simply replaying the entire recording. Wolf and Rhyne found that the temporal pattern of turn-taking was effective in identifying interesting points in the meeting record. They also suggest that using a combination of indicators such as user identification combined with a temporal threshold might make the selective review of meetings more effective. While these gaps do not have a one-to-one correspondence with pauses in speaking, the general applicability of this technique appears valid.

1.4.2 Speech Skimming and Gisting

> *A promising alternative to the fully automated recognition and understanding of speech is the detection of a limited number of key words, which would be automatically combined with linguistic and non-linguistic cues and situation knowledge in order to infer the general content or "gist" of incoming messages.*
>
> (Maksymowicz 1990, 104)

Maxemchuk suggests three techniques (after Maxemchuk 1980, 1395) for skimming speech messages:

- Text descriptors can be associated with points in a speech message. These pointers can be listed, and the speech message played back starting at a selected pointer. This is analogous to using the index in a text document to determine where to start reading.
- While playing back a speech message it is possible to jump forward or backward in the message. This is analogous to flipping through pages in a text document to determine the area of interest.
- Finally, the playback rate can be increased. When the highest playback rate is selected, not every word is intelligible; however, the meaning can generally be extracted. This is analogous to skimming through a text document to determine the areas of interest.

Several systems have been designed that attempt to obtain the gist of a recorded message (Houle 1988; Maksymowicz 1990; Rose 1991) from acoustical information. These systems use a form of keyword spotting (Wilcox 1991; Wilcox 1992a; Wilpon 1990) in conjunction with syntactic and/or timing constraints in an attempt to broadly classify a message. Similar work has recently been reported in the areas of retrieving speech documents (Glavitsch 1992) and editing applications (Wilcox 1992b).

Rose demonstrated the first complete system that takes speech messages as input, and produces as output an estimated "message class" (Rose 1991). Rose's system does not attempt to be a complete speech message understanding system that fully describes the utterance at all acoustic, syntactic and semantic levels of information. Rather, the goal is only to attempt to extract a general notion of topic or category of the input speech utterance according to a pre-defined notion of topic. The system uses a limited vocabulary Hidden Markov Model (Rabiner 1989) word spotter that provides an incomplete transcription of the speech. A second stage of processing interprets this incomplete transcription and classifies the message according to a set of pre-defined topics.

Houle et al. proposed a post-processing system for automatic gisting of speech (Houle 1988). Keyword spotting is used to detect, classify and summarize speech messages that are then used to notify an operator whenever a high-priority message arrives. They say that such a system using keyword spotting is controlled by the trade-off between the probability of detecting the spoken words and the false alarm rate. If, for example, a speaker-independent word spotting system correctly detects 80% of the individual keywords, there will be 120 false alarms per hour. With an active vocabulary of ten words in the spotting system, this would correspond to 1200 false alarms per hour, or one false alarm every three seconds. Such a system is impractical because if each keyword that was detected was used to draw the attention of the operator, the entire speech recording would effectively need to be monitored. A decreased false alarm rate was achieved by looking for two or more keywords within a short phrase window. The addition of these syntactic constraints greatly improves the effectiveness of the keyword spotting system. The other technique that they use is termed "credibility adjustment." This is effectively setting the rejection threshold of the keyword spotter to maximize the number of correct recognitions and to minimize the number of false acceptances. The application of these two kinds of back-end filters significantly reduces the number of false alarms.

It appears to be much easier to skim synthetic speech than recorded speech since the words and layout of the text (sentences, paragraphs, and formatting information) provide knowledge about the structure and content of a document. Raman (Raman 1992a, Raman 1992b) and Stevens (Edwards 1993; Stevens 1993) use such a technique for speaking documents containing mathematics based on $\TeX$ or $\LaTeX$ formatting information (Knuth 1984; Lamport 1986).

For example, as the skimming speed increases, along with raising the speed of the synthesizer, simple words such as "a" and "the" could be dropped. Wallace (Wallace 1983) and Condray (Condray 1987) applied such a technique to recorded "telegraphic speech"[3] and found that listener efficiency (the amount of information acquired per unit time) increased under such conditions. When the skimming speed of synthetic speech is increased, only selected content words or sentences could be presented. For example, perhaps only the first two sentences from each

[3]Telegraph operators often dropped common words to speed the transmission of messages.

paragraph are presented. With higher speed skimming, only the first sentence (assumed to be the "topic" sentence) would be synthesized.

Consumer products have begun to appear with rudimentary speech skimming features. Figures 1-1 and 1-2 show a telephone answering machine that incorporates time compression. The "digital message shuttle" allows the user to play voice messages at 0.7x, 1.0x, 1.3x, and 1.6x of normal speed, permits jumping back about 5 seconds within a message, and skipping forward to the next message (Sony 1993).

Fig. 1-1. A consumer answering machine with time compression.

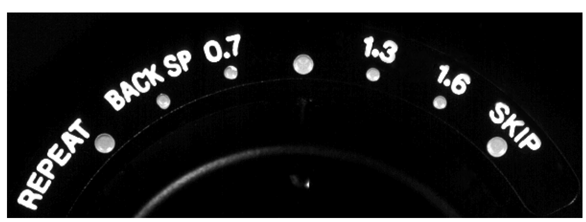


Fig. 1-2. A close-up view of the digital message shuttle.

This dissertation addresses the issues raised by these previous explorations and integrates new techniques into a single interface. This research differs from previous approaches by presenting an interactive multi-level representation based on simple speech processing and filtering of the audio signal. While existing gisting and word spotting techniques have a limited domain of applicability, the techniques presented here are invariant across all topics.

1.4.3 Speech and Auditory Interfaces

This dissertation also builds on ideas of conversational interfaces pioneered at MIT's Architecture Machine Group and Media Laboratory. Phone Slave (Schmandt 1984) and the Conversational Desktop (Schmandt 1985; Schmandt 1987) explored interactive message gathering and speech interfaces to simple databases of voice messages. Phone Slave, for example, segmented voice mail messages into five chunks[4] through an interactive dialogue with the caller.

VoiceNotes (Stifelman 1993) explores the creation and management of a self-authored database of short speech recordings. VoiceNotes investigates many of the user interface issues addressed in the Hyperspeech and SpeechSkimmer systems (chapters 2 and 5) in the context of a hand-held computer.

Resnick (Resnick 1992a; Resnick 1992b; Resnick 1992c) designed several voice bulletin board systems accessible through a touch tone interface. These systems are unique because they encourage many-to-many communication by allowing users to dynamically add voice recordings to the database over the telephone. Resnick's systems address issues of navigation among speech recordings, and include tone-based commands equivalent to "where am I" and "where can I go?" However, they require users to fill out an "audio form" to provide improved access in telephone-based information services.

These predecessor systems all structure the recorded speech information through interaction with the user, placing a burden on the creator or author of the speech data. The work presented herein automatically structures the existing recordings from information inherent in a conversational speech signal.

Muller and Daniel's description of the HyperPhone system (Muller 1990) provides a good overview of many important issues in speech-I/O hypermedia (see chapter 2). They state that navigation tends to be modeled spatially in almost any interface, and that voice navigation is particularly difficult to map into the spatial domain. HyperPhone "voice documents" are a collection of extensively interconnected fine-grained hypermedia objects that can be accessed through a speech recognition interface. The nodes contain small fragments of ASCII text to be

---

[4]Name, subject, phone number, time to call, and detailed message.

synthesized, and are connected by typed links. Hyperspeech differs from HyperPhone in that it is based on recordings of spontaneous speech rather than synthetic speech, there is no default path through the nodes, and no screen or keyboard interface of any form is provided (chapter 2).

Non-speech sounds can be used as an "audio display" for presenting information in the auditory channel (Blattner 1989; Buxton 1991). This area has been explored for applications ranging from the presentation of multidimensional data (Bly 1982) to "auditory icons" that use everyday sounds (e.g., scrapes and crashes) as feedback for actions in a graphical user interface (Gaver 1989a; Gaver 1989b; Gaver 1993).

The "human memory prosthesis" is envisioned to run on a lightweight wireless notepad-style computer (Lamming 1991). The intent is to help people remember things such as names of visitors, reconstructing past events, and locating information after it has been filed. This system is intended to gather information through active badges (Want 1992), computer workstation use, computer-based note taking, and conversations. It is noted that video is often used to record significant events such as design meetings and seminars, but that searching through this information is a tedious task. Users must play back a sufficient amount of data to reestablish context so that they can locate a small, but important, snippet of audio or video. By time-stamping the audio and video streams and correlating these with time stamps of the note taking, it is possible to quickly jump to a desired point in the audio or video stream simply by selecting a point in the hand-written notes (section 1.6.3.1).

1.5 A Taxonomy of Recorded Speech

This section broadly classifies the kinds of speech that can be captured for later playback. This taxonomy is not exhaustive, but lists the situations that are of most interest for subsequent browsing and review. Included are lists of attributes that help distinguish the classifications, and cues that can assist in segmenting and skimming the recorded speech. For example, if a user explicitly made a recording, or was present when a recording was made, the user's high-level content knowledge of the recording can assist in interactively retrieving information.

The classifications are listed roughly from hardest to easiest in terms of the ability to extract the underlying structure purely from the audio signal. Note that these classification boundaries are easily blurred. For example, there is a meeting-to-lecture continuum: parts of meetings may be more structured, as in a formal lecture, and parts of lectures may be unstructured, as in a meeting discussion.

These categories can be organized along many dimensions, such as structure, interactivity, number of people, whether self-authored, etc. Items can also be classified both within or across categories. For example, a voice mail message is from a single person, yet a collection of voice mail messages are from many people. Figure 1-3 plots these classifications as a function of number of participants and structure. The annotated taxonomy begins here:

Captured speech.

Such as a recording of an entire day's activities. This includes informal voice communication such as conversations that occur when running into someone in the hall or elevator.

- least structured
- user present
- unknown number of talkers
- variable noise
- all of the remaining items in this list

Meetings.

Including design, working, and administrative gatherings.

- more interactive than a lecture
- more talkers than a lecture
- may be a written agenda
- user may have been present or participated
- user may have taken notes

Possible cues for retrieval: who was speaking based on speaker identification, written or typed notes.

Lectures.

Including formal and informal presentations.

- typically a monologue
- organization may be more structured than a meeting
- may be a written outline or lecture notes
- user may have been present
- user may have taken notes

Possible cues for retrieval: question-and-answer period, visual aids, demonstrations, etc.

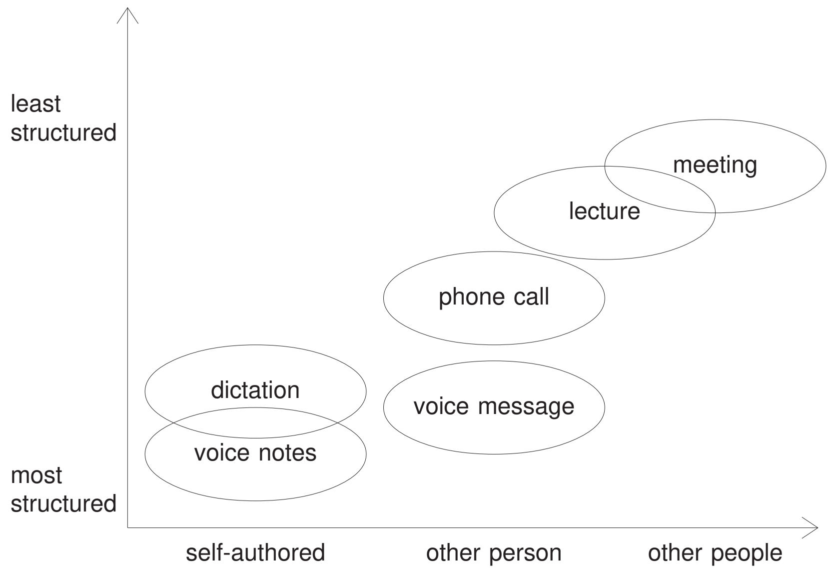


Fig. 1-3. A view of the categories in the speech taxonomy.

Personal dictation.

Such as letters or notes that would traditionally be transcribed.

- single talker
- user authored
- can be explicitly categorized by user (considered as long voice notes—see below)

Possible cues for retrieval: date, time, and place of recording.

Recorded telephone calls.

- different media than face-to-face communication
- well defined beginning and ending
- typically two talkers
- well documented speaking and hesitation characteristics (Brady 1965; Brady 1968; Brady 1969; Butterworth 1977)
- user participation in conversation
- can differentiate caller from callee (Hindus 1993)
- consistent audio quality within calls

Possible cues for retrieval: caller identification, date, time, length of call.

Voice mail.

Speech messages gathered by a computer-based answering system.

- single talker per message
- typically short
- typically contain similar types of information
- user not present
- consistent audio quality within messages

Possible cues for retrieval: caller identification, date, time, length of message.

Voice notes.
Short personal speech recordings organized by the user (Stifelman 1993). In addition to the VoiceNotes system, several other Media Laboratory projects have used small snippets of recorded speech in applications such as calendars, address books, and things-to-do lists (Schmandt 1993).

- single talker
- typically short notes
- authored by user
- consistent audio quality

Possible cues for retrieval: notes are categorized by user when authored.

Most predecessor systems rely on speech recordings that are structured in some fashion (i.e., in the lower left quadrant of figure 1-3). This dissertation attempts to segment recordings that are unstructured (i.e., in the upper right quadrant of figure 1-3).

1.6 Input (Information Gathering) Techniques

> *When the medium of communication is free-hand sketching and writing, conventional keyword searches of the meeting are not possible. The content and structure of the meeting must be inferred from other information.*
>
> (Wolf 1992, 6)

This section describes several data collection techniques that can occur when a recording is created. These data could subsequently be used as mechanisms to access and index the speech recordings.

1.6.1 Explicit

Explicit (or active) techniques require the user to manually identify interesting or important audio segments such as with button or keyboard presses (Degen 1992). Explicit techniques are uninteresting in the context of this dissertation as they burden the user during the recording process. Such techniques place an additional cognitive load on the user at record time or during authoring (see chapter 2), and do not generalize across the entire speech taxonomy. Such techniques assume that the importance and archival nature of the recording is known ahead of time. If large quantities of audio are recorded (e.g., everything that is said or heard every day), explicitly marking all recordings becomes tedious and impractical.

1.6.2 Conversational

It is also possible to use structured input techniques, or an interactive conversation with the user, to gather classification and segmentation information about a recording (see section 1.4.3). Such techniques are useful for short recordings and limited domains. However, these methods increase the complexity of creating a recording and cannot be used for all classes of recordings.

1.6.3 Implicit

> *One of the requirements in providing access to the meeting record is to do so in a way which is efficient for users who are searching the history and is not burdensome for users as they generate the meeting record.*
>
> (Wolf 1992, 1)

Implicit (or passive) techniques provide audio segmentation and search cues without requiring additional action from the user. Implicit input techniques include:

Synchronizing keyboard input with the audio recording.
Keystrokes are time-stamped and synchronized with an audio recording to provide an access mechanism into the recording (Lamming 1991). This technique is discussed further in section 1.6.3.1.

Pen or stylus synchronization with audio.
This is similar to keystroke synchronization, but uses a pen rather than a keyboard for input. Audio can thus be synchronized with handwritten notes that are recognized and turned into text, or with handwriting, figures, and diagrams that are recorded as digital “ink” or bitmaps.

CSCW keyboard synchronization.
This is a superset of keystroke or pen synchronization, but allows for multi-person input and the sharing of synchronization information between many machines.

Synchronizing an existing on-line document with audio.
The structure of an existing document, agenda, or presentation can be synchronized with button or mouse presses to provide cues to accessing chunks of recorded speech. This technique typically provides coarse-

grained chunking, but is highly correlated with topics. A technique such as this could form the basis for a CSCW application by broadcast the speaker's slides and mouse clicks to personal machines in the audience as a cue for subsequent retrieval.

Electronic white board synchronization.

The technology is similar to pen input, but is less likely to support character recognition. White board input has different social implications than keyboard or pen synchronization since the drawing space is shared rather than private.

Direction sensing microphones.

There are a variety of microphone techniques that can be used to determine where a talker is located. Each person can use a separate microphone for person identification, but this is a physical imposition on each talker. An array of microphones can be used to determine the talker's location, based on energy or phase differences between the arrival of signals at the microphone location (Flanagan 1985; Compernolle 1990). Such talker identification information can be used to narrow subsequent audio searches. Note that this technique can be used to distinguish between talkers, even if the talkers' identities are not known.

Active badge or UNIX "finger" information.

These sources can provide coarse granularity information about who is in a room, or logged on a machine (Want 1992; Manandhar 1991). These data could be combined with other information (such as direction-sensing microphones) to provide more precise information than any of the individual technologies can support.

Miscellaneous external information.

A variety of external sources such as room lights, video projectors, computer displays, or laser pointers can also be utilized for relevant synchronization information. However, it is difficult to obtain and use this information in a general way.

These techniques appear quite powerful; however, there are many times where it is useful to retrieve information from a recording created in a situation where notes were not taken, or where the other information-gathering techniques were not available. A recording, for example, may have been created under conditions where it was inconvenient or inappropriate to take notes. Additionally, something may have been said that did not seem important at the time, but on reflection, may be important to retrieve and review (Stifelman 1992a). *Therefore it is* *crucial to develop techniques that do not require input from the user during recording.*

1.6.3.1 Audio and Stroke Synchronization

Keyboard and pen synchronization techniques are the most interesting of the techniques described in section 1.6.3. They are tractable and provide fine granularity information that is readily obtained from the user. Keystrokes (or higher level constructs such as words or paragraphs) are time-stamped and synchronized with an audio recording.[5] The user's keyboard input can then be used as an access mechanism into the recording.[6]

Keyboard synchronization technology is straightforward, but may become complex with text that is manipulated and edited, or if the notes span across days or meetings. It is hypothesized that such a synchronization mechanism can be significantly more effective if it is combined with audio segmentation (section 5.9). The user's textual notes can be used as the primary means of summarizing and retrieving information; the notes provide random access to the recordings, while the recording captures *all* the spoken details, including things missed in the notes. These techniques are promising and practical, as laptop and palmtop computers are becoming increasingly common in public situations.

Note that while the intent of this technique is to not place any additional burden on the user, such a technology may change the way people work. For example, the style and quantity of notes people take in a meeting may change if they know that they can access a detailed audio recording based on their notes.

1.7 Output (Presentation) Techniques

Once recordings are created, they are processed and presented to the user. This research also explores interaction and output techniques for presenting this speech information.

[5] L. Stifelman has prototyped such a system on a Macintosh.
[6] The information that is provided by written notes is analogous to the use of keyword spotting.

1.7.1 Interaction

> *In a few seconds, or even fractions of a second, you can tell whether the sound is a news anchor person, a talk show, or music. What is really daunting is that, in the space of those few seconds, you effortlessly recognize enough about the vocal personalities and musical styles to tell whether or not you want to listen!*
>
> (Hawley 1993, 53)

Interactive user control of the audio presentation is synergistically tied to the other techniques described in this document to provide a skimming interface. User interaction is perhaps the most important and powerful of all the techniques, as it allows the *user* to filter and listen to the recordings in the most appropriate manner for a given search task (see chapters 2 and 5).

For example, most digital car radios have "scan" and "seek" buttons. Scan is automatic simply going from one station to the next. Seek allows users to go to the next station under their own control. Scan mode on a radio can be frustrating since it is simply interval-based—after roughly seven seconds, the radio jumps to the next station regardless of whether a commercial, a favorite song, or a disliked song is playing.[7] Since scan mode is automatic and hands-off, by the time one realizes that something of interest is playing, the radio has often passed the desired station. The seek command brings the listener into the loop, allowing the user to control the listening period for each station, thus producing a more desirable and efficient search. This research takes advantage of this concept, creating a closed-loop system with the user actively controlling the presentation of information.

1.7.2 Audio Presentation

There are two primary methods of presenting supplementary and navigational information in a speech-only interface: (1) the use of non-speech audio and (2) taking advantage of the spatial and perceptual processing capabilities of the human auditory system.

The use of non-speech audio cues and sound effects can be applied to this research in a variety of ways. In a speech- or sound-only interface, non-speech audio can provide terse, but informative, feedback to the user. In this research, non-speech audio is explored for providing

[7] Current radios have no cues to the semantic content of the broadcasts.

feedback to the user regarding the internal state of the system, and for navigational cues (see also Stifelman 1993).

The ability to focus one's listening attention on a single talker among a cacophony of conversations and background noise is sometimes called the "cocktail party effect." It may be possible to exploit some of the perceptual, spatial, or other characteristics of speech and audition that give humans this powerful ability to select among multiple audio streams. A spatial audio display can be used to construct a 3-D audio space of multiple simultaneous sounds external to the head (Durlach 1992; Wenzel 1988; Wenzel 1992). Such a system could be used in the context of this research to present multiple speech channels simultaneously, allowing a user to "move" between parallel speech presentations. In addition, one can take advantage of perceptually based audio streams (Bregman 1990)—speech signals can be mixed with a "primary" signal using signal processing techniques to enhance the primary sound, bringing it into the foreground of attention while still allowing the other streams to be attended to (Ludwig 1990; Cohen 1991; Cohen 1993). These areas are outside the primary research of this dissertation, but are discussed in Arons 1992b.

1.8 Summary

This chapter provides an overview of what speech skimming is, its utility, and why it is a difficult problem. A variety of related work has been reviewed, and a range of techniques that can assist in skimming speech have been presented.

Chapter 2 goes on to describe an experimental system that addresses many of these issues in the context of a hypermedia system. Subsequent chapters address the deficiencies of this experimental system and present new methods for segmenting and skimming speech recordings.

2 Hyperspeech: An Experiment in Explicit Structure

The Hyperspeech system began with a simple question: How can one navigate in a speech database using only speech? In attacking this question a variety of important issues were raised regarding structuring speech information, levels of detail, and browsing in speech user interfaces. This experiment thus motivated the remainder of this research into the issues of skimming and navigating in speech recordings.[8]

2.1 Introduction

Hyperspeech is a speech-only hypermedia application that explores issues of navigation and system architecture in an audio environment without a visual display. The system uses speech recognition to maneuver in a database of digitally recorded speech segments; synthetic speech is used for control information and user feedback.

In this prototype system, recorded audio interviews were manually segmented by topic; hypertext-style links were added to connect logically related comments and ideas. The software architecture is data-driven, with all knowledge embedded in the links and nodes, allowing the software that traverses through the network to be straightforward and concise. Several user interfaces were prototyped, emphasizing different styles of speech interaction and feedback between the user and the machine.

Interactive "hypertext" systems have been proposed for nearly half a century (Bush 1945; Nelson 1974), and realizable since the 1960's (Conklin 1987; Engelbart 1984). Attempts have continually been made to create "hypermedia" systems by integrating audio and video into traditional hypertext frameworks (Multimedia 1989; Backer 1982). Most of these systems are based on a graphical user interface paradigm using a mouse, or touch sensitive screen, to navigate through a two-dimensional space. In contrast, Hyperspeech is an application for presenting "speech

[8]This chapter is based on Arons 1991a and contains portions of Arons 1991b.

as data," allowing a user to wander through a database of recorded speech without any visual cues.

Speech interfaces *must* present information *sequentially* while visual interfaces can present information *simultaneously* (Gaver 1989a; Muller 1990). These confounding features lead to significantly different design issues when using speech (Schmandt 1989), rather than text, video, or graphics. Recorded speech cannot be manipulated, viewed, or organized on a display in the same manner as text or video images. Schematic *representations* of speech signals (e.g., waveform, energy, or magnitude displays) can be viewed in parallel and managed graphically, but the speech signals themselves cannot be listened to simultaneously (Arons 1992b). Browsing such a display is easy since it relies "on the extremely highly developed visuospatial processing of the human visual system" (Conklin 1987, 38).

Navigation in the audio domain is more difficult than in the spatial domain. Concepts such as highlighting, to-the-right-of, and menu selection must be accomplished differently in audio interfaces than in visual interfaces. For instance, one cannot "click here" in the audio world to get more information—by the time a selection is made, time has passed, and "here" no longer exists.

2.1.1 Application Areas

Applications for such a technology include the use of recorded speech, rather than text, as a brainstorming tool or personal memory aid. A Hyperspeech-like system would allow a user to create, organize, sort, and filter "audio notes" under circumstances where a traditional graphical interface would not be practical (e.g., while driving) or appropriate (e.g., for someone who is visually impaired). Speech interfaces are particularly attractive for hand-held computers that lack keyboards or large displays. Many of these ideas are discussed further in Stifelman 1992a and Stifelman 1993.

2.1.2 Related Work: Speech and Hypermedia Systems

Compared with traditional hypertext or multimedia systems, little work has been done in the area of interactive speech-only hypertext-like systems (see also section 1.4.3). Voice mail and telephone accessible databases can loosely be placed in this category; however they are far from what is considered "hypermedia." These systems generally present only a single view of the underlying data, have a limited 12-button interface, do not encourage free-form exploration of the information, and do not allow personalization of how the information is presented.

Parunak (Parunak 1989) describes five common hypertext navigational strategies in geographical terms. Hyperspeech uses a “beaten path” mechanism and typed links as additional navigational aids that reduce the complexity of the hypertext database. A beaten path mechanism (e.g., bookmarks or a back-up stack) allows a user to easily return to places already visited.

Zellweger states “Users are less likely to feel disoriented or lost when they are following a pre-defined path rather than browsing freely, and the cognitive overhead is reduced because the path either makes or narrows their choices” (Zellweger 1989, 1). Hyperspeech encourages free-form browsing, allowing users to focus on accessing information rather than navigation. Zellweger presents a path mechanism that leads a user through a hypermedia database. These paths are appropriate for scripted documents and narrations; this system focuses on conversational interactions.

IBIS has three types of nodes and a variety of link types including *questions, objects-to,* and *refers-to.* Trigg’s Textnet proposed a taxonomy of link types encapsulating ideas such as *refutation* and *support.* The system described in this chapter has two node types, and several link types similar to the argumentation links in Textnet and IBIS (Conklin 1987).

2.2 System Description

This section describes how the Hyperspeech database and links were created, and provides an overview of the hardware and software systems.

2.2.1 The Database

> *If a man can … make a better mouse-trap … the world will make a beaten path to his door.*
> R. W. Emerson

Audio interviews were conducted with five academic, research, and industrial experts in the user interface field.[9] All but one of the interviews

[9]The interviewees and their affiliations at the time were: Cecil Bloch (Somosomo Affiliates), Brenda Laurel (Telepresence Research), Marvin Minsky (MIT), Louis Weitzman (MCC Human Interface Group), and Laurie Vertelney (Apple Human Interface Group).

was conducted by telephone, since only the oral content was of interest. Note that videotaping similar interviews for a video hypermedia system would have been more expensive and difficult to schedule than telephone interviews.[10]

A short list of questions was discussed with the interviewees to help them formulate their responses before a scheduled telephone call. A telemarketing-style program then called, played recorded versions of the questions, and digitally recorded the response to each question in a different data file. Recordings were terminated without manual intervention using speech detection (see chapter 4). There were five short biographical questions (name, title, background, etc.), and three longer questions relating to the scope, present, and future of the human interface.[11] The interviews were deliberately kept short; the total time for each automated interview was roughly five minutes.

The recordings were then manually transcribed on a Sun SparcStation using a conventional text editor while simultaneously controlling audio playback with a custom-built foot pedal (figures 2-1 and 2-2). A serial mouse was built into the foot pedal, with button clicks controlling the playback of the digital recordings.

The transcripts for each question were then manually categorized into major themes (summary nodes) with supporting comments (detail nodes). Figure 2-3 is a schematic representation of the nodes in the database.[12] The starting and stopping points of the speech files corresponding to these categories were then determined with a segmentation tool. Note that most of the boundaries between segments occurred at natural pauses between phrases, rather than between words within a phrase. This attribute is of use in systems that segment speech recordings automatically (see chapter 5).

---

[10] One of the participants was in a bathtub during their telephone interview.

[11] The questions were:

1. What is the scope, or boundaries, of the human interface? What does the human interface mean to you?
2. What do you perceive as the most important human interface research issues?
3. What is the future of the human interface? Will we ever achieve “the ultimate” human interface, and if so, what will it be?

[12] The node and link images are included here with some hesitation. Such images, intended only for the author of the database, can bias a user of the system, forcing a particular spatial mapping onto the database. When the application is running, there is no visual display of any information.

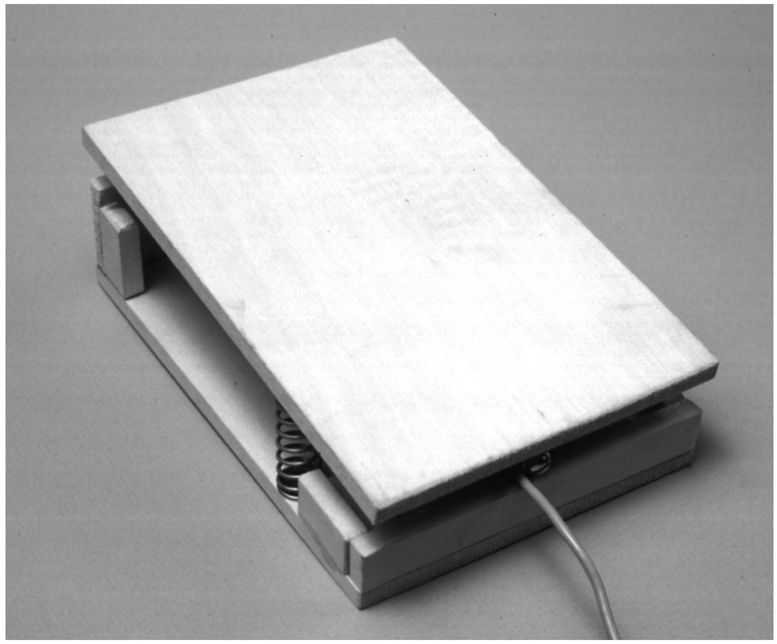


Fig. 2-1. The “footmouse” built and used for workstation-based transcription.

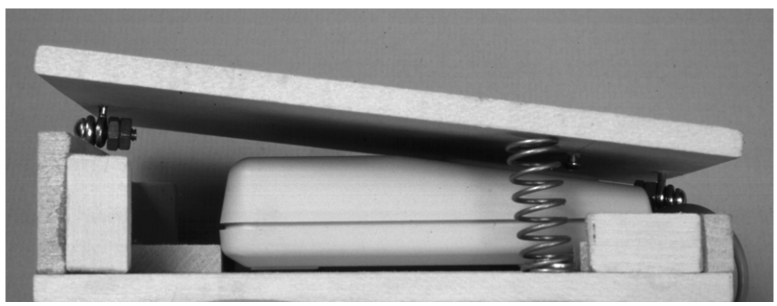


Fig. 2-2. Side view of the “footmouse.” Note the small screws used to depress the mouse buttons.

After manually analyzing printed transcripts to find interesting speech segments, a separate segmentation utility was used to determine the corresponding begin/end points in the sound file. This utility played small fragments of the recording, allowing the database author to determine segment boundaries within the sound files. Keyboard-based commands analogous to fine-, medium-, and coarse-grained cursor motions of the Emacs text editor (Stallman 1979) were used to move through the sound file and determine the proper segmentation points.[13]

[13]E.g., the forward character command (Control-F) moved forward slightly (50 ms), the forward word command (Meta-F) moved forward a small amount (250 ms), and the forward page command moved a medium amount (1 s). The corresponding backward commands also allowed movement with the recorded sound file. Other keyboard and foot-pedal commands allowed larger movements.

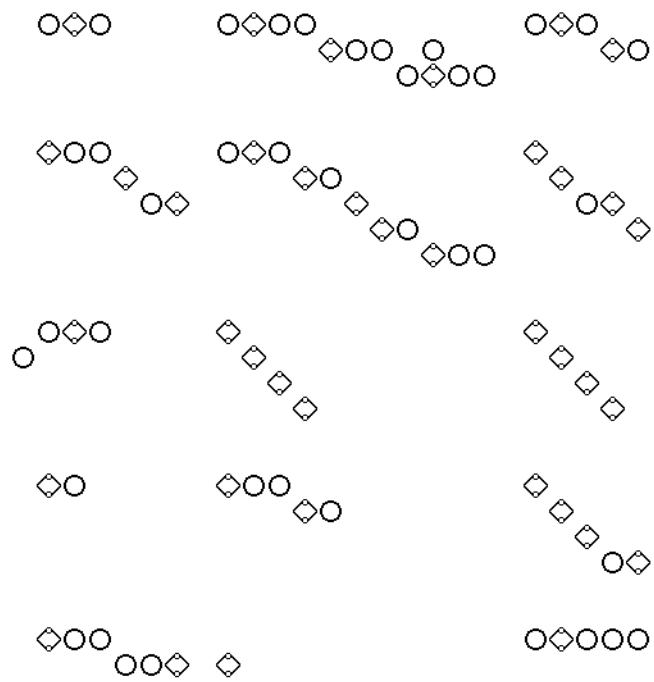


Fig. 2-3. A graphical representation of the nodes in the database. Detail nodes (circles) that are related to a summary node (diamonds) are horizontally contiguous. The three columns (narrow, wide, narrow) correspond to each of the primary questions. The five rows correspond to each of the interviewees.

Of the data gathered[14] (approximately 19 minutes of speech, including trailing silences, um's, and pauses), over 70 percent was used in the final speech database. Each of the 80 nodes[15] contains short speech segments, with a mean length of 10 seconds (*SD* = 6 seconds, maximum of 25 seconds). These brief segments parallel Muller's fine-grained hypermedia objects (Muller 1990). However, in this system each utterance represents a complete idea or thought, rather than a sentence fragment.

2.2.2 The Links

For this prototype, an X Window System-based tool designed for working with Petri nets (Thomas 1990) was used to link the nodes in the database. All the links in the system were typed according to function. Initially, a small number of *supporting* and *opposing* links between talkers were identified. For example, Minsky's comments about "implanting electrodes and other devices that can pick information out of

[14]In the remainder of the chapter, references to nodes and links do not include responses to the biographical questions.
[15]There are roughly equal numbers of summary nodes and detail nodes.

the brain and send information into the brain” are opposed to Bloch’s related view that ends “and that, frankly, makes my blood run cold.”

As the system and user interface developed, a large number of links and new link types were added (there are over 750 links in the current system). Figure 2-4 shows the links within the database. The figure also illustrates a general problem of hypermedia systems—the possibility of getting lost within a web of links. The problems of representing and manipulating a hypermedia database become much more complex in the speech domain than with traditional media.

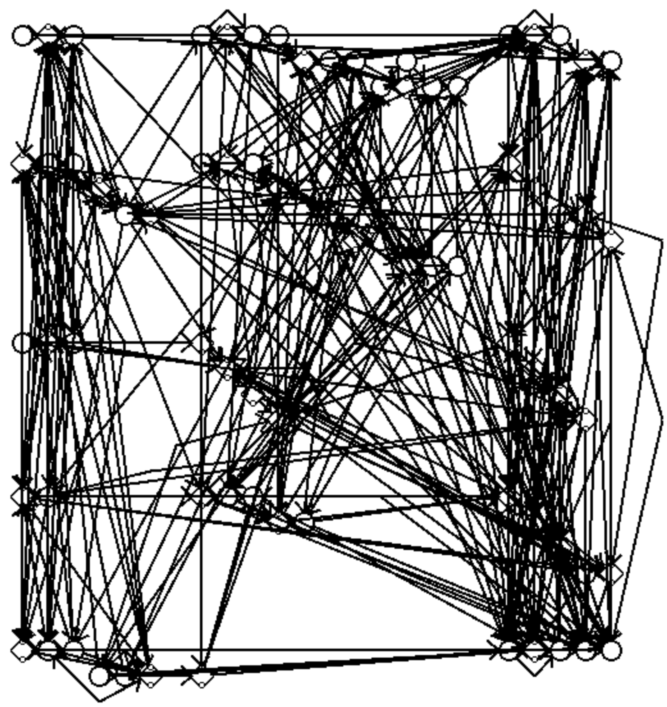


Fig. 2-4. Graphical representation of all links in the database (version 2). Note that many links are overlaid upon one another.[16]

2.2.3 Hardware Platform

The telephone interviews were gathered on a Sun 386i workstation equipped with an analog telephone interface and digitization board. The Hyperspeech system is implemented on a Sun SparcStation, using its built-in codec for playing the recorded sound segments. The telephone quality speech files are stored uncompressed (8-bit μ-law coding, 8000 samples/second). A DECtalk serial-controlled text-to-speech synthesizer is used for user feedback. The recorded and synthesized speech sounds are played over a conventional loudspeaker system (figure 2-5).

[16]A more appropriate authoring tool would provide a better layout of the links and visual differentiation of the link types.

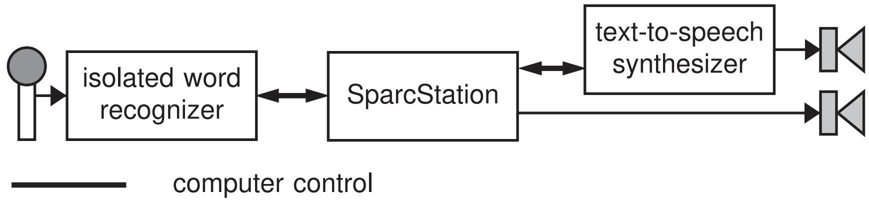


Fig. 2-5. Hyperspeech hardware configuration.

Isolated word, speaker-dependent, speech recognition is provided by a TI-Speech board in a microcomputer; this machine is used as an RS-232 controlled recognition server by the host workstation (Arons 1989; Schmandt 1988). A headset-mounted noise-canceling microphone provides the best possible recognition performance in this noisy environment with multiple sound sources (user + recordings + synthesizer).

2.2.4 Software Architecture

The software is written in C, and runs in a standard UNIX operating environment. A simple recursive stack model tracks all nodes that have been visited, and permits the user to return (pop) to a previously heard node at any time.

Because so much semantic and navigational information is embedded in the links, the software that traverses through the nodes in the database is straightforward and concise. This data-driven architecture allows the program that handles all navigation, user interaction, and feedback to be handled by approximately 300 lines of C code.[17] Note that this data-driven approach allows the researcher to scale up the size of the database without having to modify the underlying software system. This data-driven philosophy was also followed in SpeechSkimmer (see chapter 5).

2.3 User Interface Design

The Hyperspeech user interface evolved during development of the system; many improvements were made throughout an iterative design process. Some of the issues described in the following sections illustrate the differences between visual and speech interfaces, and are important design considerations for those implementing speech-based systems.

[17]Excluding extensive library routines and drivers that control the speech I/O devices.

2.3.1 Version 1

The initial system was sparsely populated with links and had a simple user interface paradigm: explicit menu control. After the sound segment associated with a node was played, a list of valid command options (links to follow) was spoken by the synthesizer. The user then uttered their selection, and the cycle was repeated.

The initial system tried to be "smart" about transitioning between the nodes. After playing the recording, if no links exited that node, the system popped the user back to the previous node, as no valid links could be followed or navigational commands issued. This automatic return-to-previous-node function was potentially several levels deep. Also, once a node had been heard, it was not mentioned in succeeding menus in order to keep the prompts as short as possible—it was (incorrectly) assumed that a user would not want to be reminded about the same node twice.

Navigation in this version was very difficult. The user was inundated with feedback from the system—the content of the recordings became lost in the noise of the long and repetitive menu prompts. The supposedly "smart" node transitions and elision of menu items brought users to unknown places, and left them stranded without landmarks because the menus were constantly changing.

2.3.2 Version 2

This section describes the current implementation of the Hyperspeech system. The most significant change from Version 1 was the addition of a variety of new link types and a large number of links.

A *name* link will transition to a node of a particular talker. For example, user input of *Minsky* causes a related comment by Marvin Minsky to be played.

Links were also added for exploring the database at three levels of detail. The *more* link allows a user to step through the database at the lowest level of detail, playing all the information from a particular talker. The *browse* link permits a user to skip ahead to the next summary node without hearing the detailed statements. This lets a user skim and get an overview of a particular talker's important ideas. The *scan*[18] command automatically jumps between the dozen or so nodes that provide a high-level overview path through the entire database, allowing a user to skim over all the recordings to find a topic of interest.

[18]In retrospect, these may not have been the most appropriate names (see section 1.1).

In order to *reduce* the amount of feedback to the user, the number of links was greatly *increased* so that a link of every type exists for each speech node. Since any link type can be followed from any node, command choices are uniform across the database, and menus are no longer needed. This is analogous to having the same graphical menu active at every node in a hypermedia interface; clicking anywhere produces a reasonable response, without having to explicitly highlight active words or screen areas. Figure 2-6 shows the vocabulary and link types of the system.

Link type	Command	Description
name	*Bloch* *Laurel* *Vertelney* *Weitzman* *Minsky*	Transition to related comments from a particular talker
dialogical	*supporting*	Transition to a node that supports this viewpoint
	opposing	Transition to a node that opposes this viewpoint
control	*more*	Transition to next detail node
	continue	Transition to next detail node (alias for *more*)
	browse	Transition to next summary node
	scan	Play path through selected summary nodes

Utilities	Command	Description
control	*return*	Pop to previous node
	repeat	Replay current node from beginning
help	*help*	Synthesize a description of current location
	options	List current valid commands
on/off	*pay attention*	Turn on speech recognizer
	stop listening	Turn off speech recognizer

Fig. 2-6. Command vocabulary of the Hyperspeech system.

A host of minor changes made the system more interactive and conversational. Since the most valuable commodity in speech systems is time rather than screen real estate, every effort was made to speed the interactions. The speech segments in the database are, by default, played back 1.25 times faster than they were recorded without a change of pitch (see chapter 3). If the *repeat* command is invoked, the node is replayed at normal speed for maximum intelligibility. The speaking rate of the synthetic speech has also been significantly increased (from the default of 180 wpm) to reduce user feedback time. Short repetitive types of feedback (e.g., direct echoing of recognized commands) are spoken at a faster rate (350 wpm) than help or navigation-related feedback (250 wpm). The output volume levels were also adjusted so that the speech recordings—the primary output of the system—are louder than the synthetic speech.

A sample interactive dialog that illustrates what the Hyperspeech system sounds like is shown in figure 2-7. See Arons 1993b for a video tape demonstrating the system in use.

Talker	Utterance	Comments
Minsky	What I think will happen over the next fifty years is we'll learn more and more about implanting electrodes, and other devices, that can pick information out of the brain and send information into the brain.	
User	*opposing*	User objects to idea, does anyone else?
Bloch	The ultimate is obviously some sort of direct coupling between the nervous system and artificial devices, and that, frankly makes my blood run cold.	
User	*browse*	Go to next summary from Bloch.
Bloch	In terms of ultimate development, I think that the thing that can be said is that it is unpredictable.	
User	*Weitzman*	What is Weitzman's view?
Weitzman	I would hope that we never do achieve the ultimate interface.	
User	*continue*	Get more information.
Weitzman	We'll always be able to improve on it, and just the fact that during the process of getting there …	
User	*help*	Interrupt to get information.
Synthesizer	This is Louie Weitzman on the future of the human interface.	
Weitzman	… we are going to learn new things and be able to see even better ways to attack the problem.	Continue playing comment.
User	*Vertelney*	What does the industrial designer think?
Vertelney	I think it's like back in the Renaissance…	
User	*return*	Not of interest. Interrupt, and go back to previous node.
Weitzman	We'll always be able to…	Weitzman again.
User	*Minsky*	What's Minsky's view of the future?
Minsky	And when it becomes smart enough we won't need the person anymore, and the interface problem will disappear.	

Fig. 2-7. A sample Hyperspeech dialog.

Explicit echoing (Hayes 1983) of recognized commands is no longer the default. However, at start-up time the system can be configured for various degrees of user feedback. Observers and first-time users of the system are often more comfortable with the interface if command echoing is turned on. As soon as a spoken command is recognized, speech output (synthesized or recorded) is immediately halted, providing crucial feedback to the user that a command was heard. The system response time is fast enough that a rejection error[19] is immediately noticeable to an experienced user. If a substitution error[20] occurs, the user can quickly engage the machine in a repair dialog (Schmandt 1986). Note that speech recognition parameters are typically set so that substitution errors are less common than rejection errors. Figure 2-8 illustrates what a repair dialog (with command echoing on) might sound like.

Talker	Utterance	Description of action
User	*Weitzman*	Desired command is spoken
Synthesizer	"supporting"	*Fast* echoing (substitution error)
Minsky	"The interfa…"	Incorrect sound is started
User	*return*	Interrupt recording, pop to previous node
User	*Weitzman*	Repeat of misrecognized command
Synthesizer	"Weitzman"	Echo of correctly recognized word
Weitzman	"I hope we never do achieve the ultimate interface…"	Desired action is taken

Fig. 2-8. An interactive repair.

2.4 Lessons Learned on Skimming and Navigating

Einstein is reported to have once said, "make everything as simple as possible, but not too simple." This idea also holds true in user interfaces, particularly those involving speech. Since time is so valuable in a speech application, every effort must be made to streamline the interactions. However, if things are made too simple, the interface also can fall apart because of the lack of identifiable landmarks. Keeping the feedback concise, or allowing various degrees of feedback to be selected, helps keep the interaction smooth and efficient. Grice's four maxims[21] about what, and how, something is said, are perhaps more applicable in machine-to-human dialogs then they are in human-to-human conversations (Grice 1975). These maxims capture many of the key ideas necessary for streamlining conversational interfaces.

[19]Rejection error: a word was spoken, but none was recognized.
[20]Substitution error: a word was spoken, but a different word was recognized.
[21]Summary of points of interest: be as informative as required, be relevant, avoid ambiguity, and be brief.

The design of this system is based on allowing the user to actively drive through the database rather than being passively chauffeured around by menus and prompts. This ability is based, in part, on having a fixed set of navigation commands that are location independent—from any location in the database, any command can be used (i.e., any link type can be followed). Note that this scheme may be difficult to implement in systems with a much larger number of nodes or link types. The total number of links is proportional to the number of nodes and the number of link types (TotalLinks = TotalNodes x LinkTypes).

To make the interactions fluent, transitions from one interaction mode to another (e.g., recognition to playback) must be designed for low system response time (Arons 1989; Schmandt 1988). Similarly, any action by the system must be easily interruptible by the user. The system should provide immediate feedback to the user that an interrupt was received; this usually takes the form of instantly halting any speech output, then executing the new command.

One general advantage of speech over other types of input modalities is that it is goal directed. A speech interface is uncluttered with artifacts of the interaction, such as menus or dialog boxes. The recognition vocabulary space is usually flat and always accessible. This is analogous to having one large pull-down menu that is always active, and contains all possible commands.

Authoring is often the most difficult part of hypermedia systems; Hyperspeech-like systems have the added complication of the serial and non-visual nature of the speech signal. Recorded speech cannot be manipulated on a display in the same manner as text or video images. Note that schematic representations of speech signals can be viewed in parallel and handled graphically, but that the speech segments represented by such a display still cannot be heard simultaneously.

One solution to managing speech recordings is to use traditional text (or hypertext) tools to manipulate transcriptions. Unfortunately, the transcription process is tedious, and the transcripts do not capture the prosody, timing, emphasis, or enthusiasm of speech that is important in a Hyperspeech-like system. Sections 2.4.1 and 2.4.2 outline ways that an audio-equipped workstation can help bridge this gap in the Hyperspeech authoring process.

2.4.1 Correlating Text with Recordings

The technology for the transcription of recorded interviews or dictation is steeped in tradition. A transcriptionist controls an analog tape machine through a foot pedal while entering text into a word processor. Modern transcribing stations have “advanced” features that can speed up or slow down the playback of recorded speech, can display the current location within the tape, and have high-speed search.

In addition to transcription, a Hyperspeech system (and many other speech-based applications) needs to accurately correlate the text with the recorded sound data. Ideally this is done automatically without explicit action by the transcriptionist—as the text is typed, a rough correspondence is made between words and points in the recorded file. An accurate one-to-one mapping between the recording and the transcription is unlikely because of the typist’s ability to listen far ahead of letters being typed at any moment (Salthouse 1984). However, even an approximate correlation is useful (Lamming 1991), allowing the hypermedia author to easily jump to the approximate sound segment and fine-tune the begin/end points to accurately match the transcription.

Once a transcript is generated, fine-grained beginning and ending points must be determined for each speech segment. A graphical editor can assist in this process by displaying the text in parallel with a visual representation of the speech signal. This allows the hypermedia author to visually locate pauses between phrases for segments of speech in the Hyperspeech database. Specialized text editors can be used for managing transcripts that have inherent structure or detailed descriptions of actions (such as data from psychological experiments that include notations for breathing, background noises, non-speech utterances, etc., see Pitman 1985).

If an accurate transcript is available, it is possible to automatically correlate the text with syllabic units detected in the recording (Hu 1987; Mermelstein 1975). For a Hyperspeech database, this type of tool would allow the hypermedia author to segment the transcripts in a text-based editor, and then create the audio file correspondences as an automated post-process. Even if the processing is not completely accurate, it would provide rough begin and end points that could be tuned manually.

2.4.2 Automated Approaches to Authoring

Unfortunately, fully automatic speaker-independent speech-to-text transcription of spontaneous speech is not practical in the near future

(Roe 1993). However, there are a variety of techniques that can be employed to completely automate the Hyperspeech authoring process (see chapter 5).

The telemarketing-style program that collected the interview database asked a series of questions that served as the foundation for the organization of the Hyperspeech database. In this prototype application, the questions were very broad, and much manual work was required to segment and link the nodes in the database. However, if the process that gathers the speech data asks very specific questions, it is possible to automatically segment and organize recorded messages by semantic content (Malone 1988; Resnick 1992b; Schmandt 1984). If the questions are properly structured (and the interviewees are cooperative), the bulk of the nodes in the Hyperspeech database can be automatically generated. This technique is particularly powerful for Hyperspeech authoring, as it not only creates the content of the database, but can link the nodes as well.

2.5 Thoughts on Future Enhancements

Hyperspeech raises as many questions as it answers. There are many improvements and extensions that can be made in terms of basic functionality and user interface design. Some of the techniques proposed in this section are intriguing, and are presented to show the untapped power of the speech communication channel.

2.5.1 Command Extensions

A variety of extensions are possible in the area of user control and feedback. Because of the difficulty of creating and locating stable landmarks in the speech domain, it is desirable to be able to dynamically add personalized bookmarks (the need for this feature reappears in section 5.10.13). While listening to a particular sound segment the user might say "*bookmark:* hand-held computers," creating a new method of accessing that particular node. Note that the name of the bookmark does not have to be recognized by the computer the first time it is used. Instead, after recognizing the key phrase *bookmark,* a new recognizer template is trained on-the-fly with the utterance following the key phrase (Stifelman 1992a; Stifelman 1993). A subsequent "*go to:* hand-held computers" command, will take the user back to the appropriate node and sound segment.

Besides adding links, it is desirable to dynamically extend the database by adding new nodes. For example, using a scheme similar to that of adding bookmarks, the user can record a new node by saying[22] "*add supporting:* conversational interfaces will be the most important development in the next 20 years." This creates new *supporting* and *name* links, as well as a node representing the newly recorded speech segment.[23] A final variant of this technique is to dynamically generate new link types. For example, a command of the form "*link to:* hand-held computers, *call it:* product idea" would create a *product idea* link between the bookmark and the currently active node.[24]

There are many speech commands that can be added to allow easier navigation and browsing of the speech data. For example, a command of the form *"Laurel on Research"* would jump to a particular talker's comments on a given topic. It is also possible to add commands, or command modifiers, that allow automated cross-sectional views or summary paths through the database. Command such as *"play all Minsky"* or *"play all future"* would play all of Minsky's comments or all comments about the future of the human interface. It may also be possible to generate on-the-fly arguments between the interviewees. A command such as *"contrast Bloch and Vertelney on the scope of the human interface"* could create a path through the database simulating a debate.

2.5.2 Audio Effects

Audio cues can provide an indication of the length of a given utterance, a feature particularly useful if there is a wide range of recording lengths. Some voice mail systems, for example, inform the user "this is a long message" before playing a long-winded recording[25] (Stifelman 1991). In Hyperspeech, where playing sounds is the fundamental task of the system, a more efficient (less verbose) form of length indication is desired. For example, playing a short (perhaps 50 millisecond) high pitched tone might indicate a brief recording, while a longer (250 ms) low tone may suggest a lengthy recording (Bly 1982; Buxton 1991). Doppler effect frequency shifts of a speech segment can also suggest that

[22]An isolated word recognizer can be trained with short utterances (e.g., "add supporting") in addition to single words. Some of the examples presented in this section, however, would be better handled by a continuous speech recognizer.
[23]One complication of this design is that it may create nodes that are under populated with links. This may not present a problem if such nodes are sparsely distributed throughout the database.
[24]Many links can be generated, including *product idea* and *name* links in both directions.
[25]Note that it is counterproductive to say "this is a short message."

the user is approaching, or passing, a Hyperspeech branch that exists only in time.

2.6 Summary

The Hyperspeech system provides an introduction to the possibilities of constructing speech-only hypermedia environments and interactively skimming speech recordings. An important implication of Hyperspeech is that it is significantly different to create and navigate in speech-only hypermedia than it is to augment, or use, visually based hypermedia with speech (see also Mullins 1993).

It is difficult to capture the interactive conversational aspects of the system by reading a written description. Most people who have heard this interface have found it striking, and its implications far reaching. One user of the system felt that they were “creating artificial conversations between Minsky and Laurel” and that the ability to stage such conversations was very powerful.

Many of the ideas developed in the Hyperspeech system, such as different levels of representation, interactive control, and the importance of time in speech interfaces, can be applied to create a more general form of interaction with unstructured speech data. During the Hyperspeech authoring process, it became painfully clear that continued development of such a system would require significantly better, or radically different, authoring tools and techniques. The remainder of this dissertation addresses these issues.

3 Time Compression of Speech

That is to say, he can listen faster than an experienced speaker can talk.
(Smith 1970, 219)

Hyperspeech (chapter 2) and previous interactive systems have demonstrated the importance of managing time in speech-based systems. This chapter investigates methods for removing redundancies in speech, to allow recordings to be played back in less time than it took to create them. The ideas presented in this chapter are a crucial component of the SpeechSkimmer system described in chapter 5.

A variety of techniques for time compressing speech have been developed over the last four decades. This chapter contains a review of the literature on methods for time compressing speech, including related perceptual studies of intelligibility and comprehension.[26]

3.1 Introduction

Time-compressed speech is also referred to as accelerated, compressed, time-scale modified, sped-up, rate-converted, or time-altered speech. “Time-scale modified” is often used in the digital signal processing literature; “time-compressed” or “accelerated” is often used in the psychology literature. Time-compressed is used here instead of time-scale modified since the goal of this research is to make things faster, rather than slow things down.

The primary motivation for time-compressed speech is to reduce the time needed for a user to listen to a message—to increase the communication capacity of the ear. A secondary motivation is that of data reduction—to save storage space and transmission bandwidth for speech messages.

Time-compressed speech can be used in a variety of application areas including teaching, aids to the disabled, and human-computer interfaces. Studies have indicated that listening twice to teaching materials that have

[26]This chapter is based on Arons 1992a.

been speeded up by a factor of two is more effective than listening to them once at normal speed (Sticht 1969). Time-compressed speech has been used to speed up message presentation in voice mail systems (Maxemchuk 1980; Hejna 1990), and in aids for the visually impaired. Speech can be slowed for learning languages, or for the hearing impaired. Time compression techniques have also been used in speech recognition systems to time normalize input utterances to a standard length (Malah 1979; Watanabe 1992).

While the utility of time compressing recordings is generally recognized, surprisingly, its use has not become pervasive. Rippey performed an informal study on users of a time compression tape player installed in a university library. Virtually all the comments on the system were positive, and the librarians reported that the speech compressor was the most popular piece of equipment in the library (Rippey 1975).

The lack of commercial acceptance of time-compressed speech is partly because of the cost of compression devices and the quality of the reproduced speech, but is also attributable to the lack of user control. Traditionally, recordings were reproduced at fixed compression ratios where:

> the rate of listening is completely paced by the recording and is not controllable by the listener. Consequently, the listener cannot scan or skip sections of the recording in the same manner as scanning printed text, nor can the listener slow down difficult-to-understand portions of the recording. (Portnoff 1978, 10)

3.1.1 Time Compression Considerations

The techniques presented in this chapter can be applied to a wide range of recordings, and used under disparate listening conditions. The items listed in this section should be kept in mind while reading the remainder of this document, and while designing time compression techniques appropriate for a given interactive speech application.

There are three variables that can be studied in compressed speech (Duker 1974):

- The type of speech material to be compressed: content, language, background noise, etc.
- The process of compression: algorithm, monophonic or stereophonic presentation, etc.
- The listener: prior training, intelligence, listening task, etc.

Other related factors come into play in the context of integrating speech into computer workstations or hand-held computers:

- Is the material familiar or self-authored, or is it unfamiliar to the listener?
- Does the recorded material consist of many short items, or large unsegmented chunks of speech?
- Is the user quickly browsing or listening for maximum comprehension?

3.1.2 A Note on Compression Figures

There are several ways to express the amount of compression produced by the techniques described in this document. The most common figure in the literature is the compression percentage.[27] A compression of 50% corresponds to a factor of two increase in speed (2x), halving the time required to listen. A compression of 20% corresponds to a factor of five increase in speed. These numbers are most easily thought of as the total reduction in time or data.

3.2 General Time compression Techniques

A variety of techniques for increasing the playback speed of speech are described briefly in the following sections (most of these methods also work for slowing down speech). Note that these techniques are primarily concerned with reproducing the entire recording, not skimming portions of the signal. Much of the research summarized here was performed between the mid-1950's and the mid-1970's, often in the context of developing accelerated teaching techniques, or aids for the visually impaired.

3.2.1 Speaking Rapidly

The normal English speaking rate is in the range of 130–200 words per minute (wpm). When speaking fast, talkers unintentionally change relative attributes of their speech such as pause durations, consonant-vowel duration, etc. Talkers can only compress their speech to about 70% because of physiological limitations (Beasley 1976).[28]

---

[27] An attempt has been made to present all numbers quoted from the literature in this format.

[28] However, according to the Guinness Book of World Records, John Moschitta has been clocked speaking at a rate of 586 wpm. Mr. Moschitta is best known for his roles as the fast-talking businessman in Federal Express television commercials.

3.2.2 Speed Changing

Speed changing is analogous to playing a tape recorder at a faster (or slower) speed. This method can be replicated digitally by changing the sampling rate during the playback of a sound. Techniques such as these are undesirable since they produce a frequency shift[29] proportional to the change in playback speed, causing a decrease in intelligibility.

3.2.3 Speech Synthesis

With purely synthetic speech (Klatt 1987) it is possible to generate speech at a variety of word rates. Current text-to-speech synthesizers can produce speech at rates up to 350–550 wpm. This is typically done by selectively reducing the phoneme and silence durations. This technique is useful, particularly in aids for the disabled, but is not relevant to recorded speech. Note that these maximum speech rates are higher than many of the figures cited in the remainder of this chapter because of special requests by members of the blind community.

3.2.4 Vocoding

Vocoders (voice coders) that extract pitch and voicing information can be used to time compress speech. For example, if a vocoder that extracts speech features every 20 ms is used to drive a decoder that expects speech data every 10 ms, the speech will be compressed by 50%. Most vocoding efforts, however, have focused on bandwidth reduction rather than on naturalness and high speech quality. The phase vocoder (section 3.4.2) is a high quality exception.

3.2.5 Pause Removal

A variety of techniques can be used to find pauses (hesitations) in speech and remove them since they contain no lexical information. The resulting speech is “natural, but many people find it exhausting to listen to because the speaker never pauses for breath” (Neuburg 1978, 624).

The simplest methods involve the use of energy or average magnitude measurements combined with time thresholds; other metrics include zero crossing rate measurements, LPC parameters, etc. A variety of speech and background noise detection techniques are reviewed in detail in chapter 4.

[29]Causing the talker to sound like Mickey Mouse or the “Chipmunks.”

3.3 Time Domain Techniques

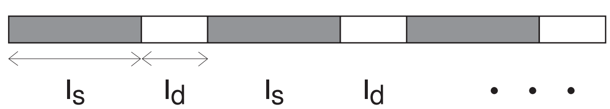
The most practical time compression techniques work in the time domain and are based on removing redundant information from the speech signal. The most common of these techniques are discussed in this section.

3.3.1 Sampling

The basis of much of the research in time-compressed speech was established in 1950 by Miller and Licklider's experiments that demonstrated the temporal redundancy of speech. The motivation for this work was to increase communication channel capacity by switching the speech on and off at regular intervals so the channel could be used for another transmission (figures 3-1 and 3-2B). It was established that if interruptions were made at frequent intervals, large portions of a message could be deleted without affecting intelligibility (Miller 1950).

Other researchers concluded that listening time could be saved by abutting the interrupted speech segments. This was first done by Garvey who manually spliced audio tape segments together (Garvey 1953a; Garvey 1953b), then by Fairbanks with a modified tape recorder with four rotating pickup heads (Fairbanks 1954).

The bulk of literature involving the intelligibility and comprehension of time-compressed speech is based on such electromechanical tape recorders. In the Fairbanks (or sampling) technique, segments of the speech signal are alternatively discarded and retained, as shown in figure 3-2C. This has traditionally been done isochronously—at constant sampling intervals without regard to the content of the signal.



sampling interval I_s
discard interval I_d
time compression ratio $R_c = I_d/(I_d+I_s)$

Fig. 3-1. Sampling terminology (after Fairbanks 1957).

Word intelligibility decreases if I_d is too large or too small. Portnoff notes that the duration of each sampling interval should be at least as long as one pitch period (e.g., >~15 ms), but should also be shorter than the length of a phoneme (Portnoff 1981). Although computationally simple, such time-domain techniques introduce discontinuities at the interval boundaries that are perceived as “burbling” distortion and general signal degradation.

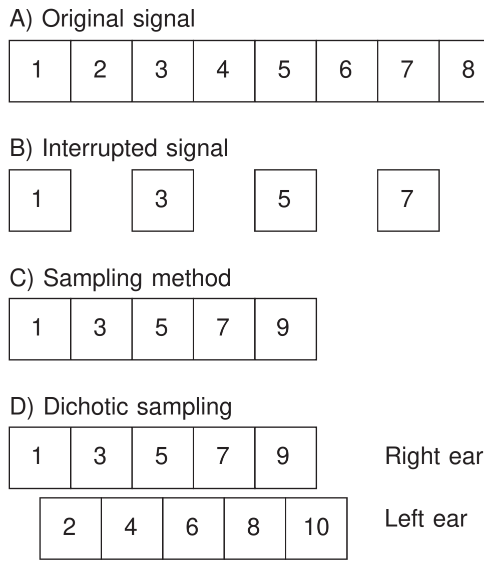


Fig. 3-2. (A) is the original signal; the numbered regions represent short (e.g., 50 ms) segments. Signal (B) is still intelligible. For a 2x speed increase using the sampling method (C), every other chunk of speech from the original signal is discarded. The same technique is used for dichotic presentation, but different segments are played to each ear (D).

It has been noted that some form of windowing function or digital smoothing at the junctions of the abutted segments will improve the audio quality. The “braided-speech” method continually blended adjacent segments with linear fades, rather than abutting segments (Quereshi 1974). Lee describes two digital electronic implementations of the sampling technique (Lee 1972), and discusses the problems of discontinuities when segments are simply abutted together.

3.3.2 Sampling with Dichotic Presentation

> *One of the most striking facts about our ears is that we have two of them—and yet we hear one acoustic world; only one voice per speaker.*
> (Cherry 1954, 554)

Sampling with dichotic[30] presentation is a variant of the sampling method that takes advantage of the auditory system’s ability to integrate information from both ears (figure 3-2D). It improves on the sampling method by playing the standard sampled signal to one ear and the

[30]*Dichotic* means a different signal is presented to each ear; *diotic* means the same signal is presented to both ears; *monotic* means a signal is presented to only one ear.

"discarded" material to the other ear[31] (Scott 1967, summarized in Orr 1971). Under this dichotic condition, where different signals are presented to each ear over headphones, both intelligibility and comprehension increase. Most subjects also prefer this technique to a diotic presentation of a conventionally sampled signal. Listeners initially reported a switching of attention between ears, but they quickly adjusted to this unusual sensation. Note that for compression ratios up to 50%, the two signals to the ears contain common information. For compression greater than 50% some information is necessarily lost.

3.3.3 Selective Sampling

The basic sampling technique periodically removes pieces of the speech waveform without regard to whether it contains unique or redundant speech information. David and McDonald demonstrated a bandwidth reduction technique in 1956 that selectively removed redundant pitch periods from speech signals (David 1956). Scott applied the same ideas to time compression, setting the sampling and discard intervals to be synchronous with the pitch periods of the speech. Discontinuities in the time compressed signal were reduced, and intelligibility increased (Scott 1972). Neuburg developed a similar technique in which intervals equal to the pitch period were discarded (but not synchronous with the pitch pulses). Finding the pitch pulses is hard (Hess 1983), yet estimating the pitch period is much easier, even in noisy speech (Neuburg 1978).

Since frequency-domain properties are expensive to compute, it has been suggested that easy-to-extract time-domain features can be used to segment speech into transitional and sustained segments. For example, simple amplitude and zero crossing measurements for 10 ms frames can be used to group adjacent frames for similarity—redundant frames can then be selectively removed (Quereshi 1974). Toong selectively deleted 50–90% of vowels, up to 50% of consonants and fricatives, and up to 100% of pauses (Toong 1974). However, it was found that complete elimination of pauses was undesirable (see also section 3.7.4). Portnoff summarized these findings:

> The most popular refinement of the Fairbanks technique is pitch-synchronous implementation. Specifically, for portions of speech that are voiced, the sections of speech that are repeated or discarded correspond to pitch periods. Although this scheme produces more intelligible speech than the basic asynchronous pitch-independent method, errors in pitch marking and voiced-unvoiced decisions introduce objectionable artifacts… Perhaps the most successful variant

[31] Often with a delay of half of the discard interval.

> of the Fairbanks method is that recently proposed by Neuburg. This method uses a crude pitch detector, followed by an algorithm that repeats or discards sections of the speech equal in length to the average pitch period then smooths together the edges of the sections that are retained. Because the method is not pitch synchronous, and, therefore, does not require pitch marking, it is more robust than pitch-synchronous implementations, yet much higher quality than pitch-independent methods. (Portnoff 1978, 12)

3.3.4 Synchronized Overlap Add Method

The synchronized overlap add method (SOLA) first described by Roucos and Wilgus (Roucos 1985) has recently become popular in computer-based systems. It is a variant of a Fourier-based algorithm described by Griffin and Lim (Griffin 1984), but is optimized to eliminate the need for an iterative solution. "Of all time scale modification methods proposed, SOLA appears to be the simplest computationally, and therefore most appropriate for real-time applications" (Wayman 1989, 714). Conceptually, the SOLA method (figure 3-3) consists of shifting the beginning of a new speech segment over the end of the preceding segment to find the point of highest cross-correlation (i.e., maximum similarity). Once this point is found, the frames are overlapped and averaged together, as in the sampling method. SOLA provides a locally optimal match between successive frames (the technique does not attempt to provide global optimality). The shifts do not accumulate since the target position of a window is independent of any previous shifts (Hejna 1990).

Combining the frames in this manner tends to preserve the time-dependent pitch, magnitude, and phase of a signal. The SOLA method is simple and effective as it does not require pitch extraction, frequency-domain calculations, or phase unwrapping, and is non-iterative (Makhoul 1986). The SOLA technique can be considered a type of selective sampling that effectively removes redundant pitch periods.

A windowing function can be used with this technique to smooth between segments, producing significantly fewer artifacts than traditional sampling techniques. Makhoul used both linear and raised cosine functions for averaging windows, and found the simpler linear function sufficient (Makhoul 1986). The SOLA algorithm is robust in the presence of noise, and can improve the signal-to-noise ratio of noisy speech since the cross-correlation tends to align periodic features (i.e., speech) in the signal (Wayman 1988; Wayman 1989).

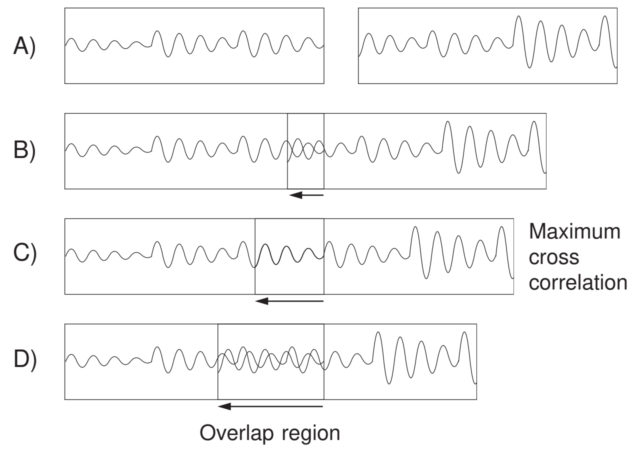


Fig. 3-3. SOLA: shifting the two speech segments (as in figure 3-2) to find the maximum cross correlation. The maximum similarity occurs in case C, eliminating a pitch period.

Several improvements to the SOLA method have been suggested that offer improved computational efficiency, or increased robustness in data compression applications (Makhoul 1986; Wayman 1988; Wayman 1989; Hardam 1990; Hejna 1990). Hejna, in particular, provides a detailed description of SOLA, including an analysis of the interactions of various parameters used in the algorithm. Hejna states:

> Ideally the modification should remove an integer multiple of the local pitch period. These deletions should be distributed evenly throughout the segment, and to preserve intelligibility, no phoneme should be completely removed. (Hejna 1990, 2)

3.4 Frequency Domain Techniques

In addition to the frequency domain methods outlined in this section, there are a variety of other frequency-based techniques that can be used for time compressing speech (e.g., McAulay 1986; Quatieri 1986).

3.4.1 Harmonic Compression

Harmonic compression involves the use of a fine-tuned (typically analog) filter bank. The energy outputs of the filters are used to drive filters at half of the original frequency. A tape of the output of this system is then played on a tape recorder at twice normal speed. The compression ratio of this frequency domain technique was fixed, and was being developed before it was practical to use digital computers for time compression.

Malah describes time-domain harmonic scaling that requires pitch estimation, is pitch synchronous, and can only accommodate certain compression ratios (Malah 1979; Lim 1983).

3.4.2 Phase Vocoding

A vocoder that maintains phase (Dolson 1986) can be used for high quality time compression. A “phase vocoder” can be interpreted as a filter bank and thus is similar to the harmonic compressor. A phase vocoder is, however, significantly more complex because calculations are done in the frequency domain, and the phase of the original signal must be reconstructed.

Portnoff developed a system for time-scale modification of speech based on short-time Fourier analysis (Portnoff 1981). The system provided high quality compression of up to 33% (3x) while retaining the natural quality and speaker-dependent features of the speech. The resulting signals were free from artifacts such as glitches, burbles, and reverberations typically found in time-domain methods of compression such as sampling.

Phase vocoding techniques are more accurate than time domain techniques, but are an order of magnitude more computationally complex because Fourier analysis is required. Dolson says, “A number of time-domain procedures … can be employed at substantially less computational expense. But from a standpoint of fidelity (i.e., the relative absence of objectionable artifacts), the phase vocoder is by far the most desirable” (Dolson 1986, 23). The phase vocoder is particularly good at slowing speech down to hear features that cannot be heard at normal speed—such features are typically lost using the time-domain techniques described in section 3.3.

3.5 Tools for Exploring the Sampling Technique

A variety of software tools and utilities were built for investigating variants of the sampling method (sections 3.3.1 and 3.3.2) and new ways to combine time compression techniques (section 3.6) for the SpeechSkimmer system. Figure 3-4 shows some of the parameters available in the sampling tool. Additional tools enabled speech or background noise segments to be extracted from a sound file, two files to be interleaved for dichotic presentation, SOLA time compression, etc.

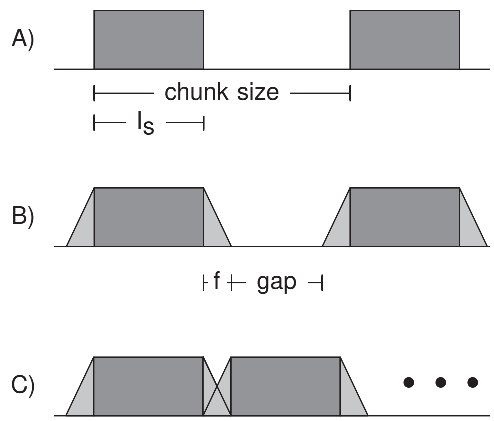


Fig. 3-4. Parameters used in the sampling tool. In (A) the sampling interval, l_S, is specified as a fraction of the chunk size. In (B), the length of the linear fade, f, is specified at the chunk boundaries. The gap length can be set to allow time between fades (in B), to abut the fade segments, or to overlap the fades for a linear cross fade (in C).

These software tools permitted the rapid exploration of combined and novel time compression techniques (sections 3.6.2 and 3.6.3). For example, the speech segments could be extracted from a file with different amounts of the background noise inserted between segments. The length of the background noise segments can be a fraction of the actual noise between speech segments, set to a predefined length, or linearly interpolated between two set values based on the actual length of the pauses. These explorations led to the time compression and pause removal parameters used in the final SpeechSkimmer design.

3.6 Combined Time Compression Techniques

The time compression techniques described earlier in this chapter can be mixed and matched in a variety of ways. Such combined methods can provide a variety of signal characteristics and a range of compression ratios.

3.6.1 Pause Removal and Sampling

Maxemchuk found that eliminating every other non-silent block (1/16 second) produced “extremely choppy and virtually unintelligible playback” (Maxemchuk 1980, 1392). Eliminating intervals with less energy than the short-term average (and no more than one in a row), produced distorted but intelligible speech. This technique produced compressions of 33 to 50 percent. Maxemchuk says that this technique:

> has the characteristic that those words which the speaker considered to be most important and spoke louder were

> virtually undistorted, whereas those words that were spoken softly are shortened. After a few seconds of listening to this type of speech, listeners appear to be able to infer the distorted words and obtain the meaning of the message. (Maxemchuk 1980, 1393)

Maxemchuk believes such a technique would be:

> useful for users of a message system to scan a large number of messages and determine which they wish to listen to more carefully or for users of a dictation system to scan a long document to determine the areas they wish to edit. (Maxemchuk 1980, 1393)

Pause compression and sampling can be combined in several ways. Pauses can first be removed from a signal that is then sampled. Alternatively, the output of a speech detector can be used to set boundaries for sampling, producing a selective sampling technique. Note that using pauses to find discard intervals eliminates the need for a windowing function to smooth (de-glitch) the sound at the boundaries of the sampled intervals.

3.6.2 Silence Removal and SOLA

On the surface it appears that removing silences and time compressing speech with SOLA should be linearly independent, and could thus be performed in any order. In practice there are some minor differences, because the SOLA algorithm makes assumptions about the properties of the speech signal. Informal tests found a slight improvement in speech quality by applying the SOLA algorithm before removing silences. Note that the silence removal timing parameters must be modified under these conditions. For example, with speech sped up by a factor of two, the silence removal timing thresholds must be cut in half. This combined technique is effective, and can produce a fast and dense speech stream. Note that silence periods can be selectively retained or shortened, rather than simply removed to provide the listener with cognitive processing time.

3.6.3 Dichotic SOLA Presentation

A sampled signal compressed by 2x can be presented dichotically so that exactly half the signal is presented to one ear, while the remainder of the signal is presented to the other ear. Generating such a lossless dichotic presentation is difficult with the SOLA method because the segments of speech are shifted relative to one another to find the point of maximum cross correlation. However, by choosing two starting points in the speech data carefully (based on the parameters used in the SOLA algorithm), it is possible to maximize the difference between the signals presented to the two ears. This technique has been informally found to be effective since it combines the high quality sounds produced with the SOLA algorithm with the advantages of dichotic presentation.

3.7 Perception of Time-Compressed Speech

There has been a significant amount of perceptual work performed in the areas of intelligibility and comprehension of time-compressed speech. Much of this research is summarized in (Beasley 1976; Foulke 1969; Foulke 1971).

3.7.1 Intelligibility versus Comprehension

"Intelligibility" usually refers to the ability to identify isolated words. Depending on the type of experiment, such words may either be selected from a closed set or written down (or shadowed) by the subject from an open-ended set. "Comprehension" refers to the understanding of the content of the material. This is usually tested by asking questions about a passage of recorded material.

Intelligibility is generally more resistant to degradation as a function of time compression than is comprehension (Gerber 1974). Early studies showed that single well-learned phonetically balanced words could remain intelligible with a 10–15% compression (10x normal speed), while connected speech remains comprehensible to a 50% compression (2x normal speed).

> If speech, when accelerated, remains comprehensible the savings in listening time should be an important consideration in situations in which extensive reliance is placed on aural communication. However, current data suggest that although individual words and short phrases may remain intelligible after considerable compression by the right method, when these words are combined to form meaningful sequences that exceed the immediate memory span for heard words, as in a listening selection, comprehension begins to deteriorate at a much lower compression. (Foulke 1971, 79)

3.7.2 Limits of Compression

There are some practical limitations on the maximum amount that a speech signal can be compressed. Portnoff notes that arbitrarily high compression ratios are not physically reasonable. He considers, for example, a voiced phoneme containing four pitch periods. Greater than 25% compression reduces this phoneme to less than one pitch period, destroying its periodic character. Thus, high compression ratios are expected to produce speech with a rough quality and low intelligibility (Portnoff 1981).

The "dichotic advantage" (section 3.3.2) is maintained for compression ratios of up to 33%. For discard intervals between 40–70 ms, dichotic intelligibility was consistently higher than diotic (same signal to both ears) intelligibility (Gerber 1977). A dichotic discard interval of 40–50 ms was found to have the highest intelligibility (40 ms was described as the "optimum interval" in another study, see Gerber 1974; earlier studies suggest that a shorter interval of 18–25 ms may be better for *diotic* speech, see Beasley 1976).

Gerber showed that 50% compression presented diotically was significantly better than 25% compression presented dichotically, even though the information quantity of the presentations was the same. These and other data provide conclusive evidence that 25% compression is too fast for the information to be processed by the auditory system. The loss of intelligibility, however, is not due to the loss of information because of the compression process (Gerber 1974).

Foulke reported that comprehension declines slowly up to a word rate of 275 wpm, but more rapidly beyond that point (Foulke 1969). The decline in comprehension was not attributable to intelligibility alone, but was related to a processing overload of short-term memory. Recent experiments with French have shown that intelligibility and comprehension do not significantly decay until a high rate (300 wpm) is reached (Richaume 1988).

Note that in much of the literature the limiting factor that is often cited is word rate, not compression ratios. The compression required to boost the speech rate to 275 words per minute is both talker- and context-dependent (e.g., read speech is typically faster than spontaneous speech).

Foulke and Sticht permitted sighted college students to select a preferred degree of time compression for speech spoken at an original rate of 175 wpm. The mean preferred compression was 82%, corresponding to a word rate of 212 wpm. For blind subjects it was observed that 64–75% time compression and word rates of 236–275 words per minute were preferred. These data suggest that blind subjects will trade increased effort in listening to speech for a greater information rate and time savings (Zemlin 1968).

In another study (Heiman 1986), comprehension of interrupted speech (see section 3.3.1) was good, probably because the temporal duration of the original speech signal was preserved, providing ample time for subjects to attempt to process each word. Compression requires that each portion of speech be perceived in less time than normal. However, each unit of speech is presented in a less redundant context, so that more time per unit is required. Based on a large body of work in compressed speech, Heiman et al. suggest that 50% compression removes virtually all redundant information. With greater than 50% compression, critical non-redundant information is also lost. They conclude that the compression ratio rather than word rate is the crucial parameter, because greater than 50% compression presents too little of the signal in too little time for enough words to be accurately perceived. They believe that the 275 wpm rate is of little significance, but that compression and its underlying temporal interruptions decrease word intelligibility that results in decreased comprehension.

3.7.3 Training Effects

As with other cognitive activities, such as listening to synthetic speech, exposure to time-compressed speech increases both intelligibility and comprehension. There is a novelty in listening to time-compressed speech for the first time that is quickly overcome with experience.

Even naive listeners can tolerate compressions of up to 50%, and with 8–10 hours of training, substantially higher speeds are possible (Orr 1965). Orr hypothesizes that "the review of previously presented material could be more efficiently accomplished by means of compressed speech; the entire lecture, complete with the instructor's intonation and emphasis, might be re-presented at high speed as a review" (Orr 1965, 156). Voor found that practice increased comprehension of rapid speech, and that adaptation time was short—minutes rather than hours (Voor 1965).

Beasley reports on an informal basis that following a 30 minute or so exposure to compressed speech, *listeners become uncomfortable if they are forced to return to the normal rate of presentation* (Beasley 1976). Beasley also reports on a controlled experiment extending over a six-week period that found subjects' listening rate preference shifted to faster rates after exposure to compressed speech.

3.7.4 The Importance of Pauses

> *Well-timed silence hath more eloquence than speech.*
>
> Martin Farquhar Tupper
> *Proverbial Philosophy,* 1838–42

> Just as pauses are critical for the speaker in facilitating fluent and complex speech, so are they crucial for the listener in enabling him to understand and keep pace with the utterance. (Reich 1980, 388)

> the debilitating effects of compressed speech are due as much to depriving listeners of ordinarily available processing time, as to degradation of the speech signal itself. (Wingfield 1980, 100)

It may not be desirable to completely remove pauses, as they often provide important semantic and syntactic cues. Wingfield found that with normal prosody, intelligibility was higher for syntactic segmentation (inserting silences after major clause and sentence boundaries) than for periodic segmentation (inserting 3 s pauses after every eighth word). Wingfield says that "time restoration, especially at high compression ratios, will facilitate intelligibility primarily to the extent that these presumed processing intervals coincide with the linguistic structure of the speech materials" (Wingfield 1984, 133)

In another experiment, subjects were allowed to stop time-compressed recordings at any point, and were instructed to repeat what they had heard (Wingfield 1980). It was found that the average reduction in selected segment duration was almost exactly proportional to the increase in the speech rate. For example, the mean segment duration for the normal speech was 3 seconds, while the chosen segment duration of speech compressed 60% was 1.7 seconds. Wingfield found that:

> while time and/or capacity must clearly exist as limiting factors to a theoretical maximum segment size which could be held [in short-term memory] for analysis, speech content as defined by syntactic structure, is a better predictor of subjects' segmentation intervals than either elapsed time or simple number of words per segment. This latter finding is robust, with the listeners' relative use of the [syntactic] boundaries remaining virtually unaffected by increasing speech rate. (Wingfield 1980, 100)

In the perception of normal speech, it has been found that pauses exerted a considerable effect on the speed and accuracy with which sentences were recalled, particularly under conditions of cognitive complexity (Reich 1980). Pauses, however, are only useful when they occur between clauses within sentences—pauses within clauses are disrupting. When a 330 ms pause was inserted ungrammatically, response time for a particular task was increased by 2 seconds. Pauses suggest the boundaries of material to be analyzed, and provide vital cognitive processing time.

Maxemchuk found that eliminating hesitation intervals decreased playback time of recorded speech with compression ratios of 50 to 75 percent depending on the talker and material. In his system a 1/8 second pause is inserted whenever a pause greater or equal to 1 second occurred in a message. This appeared to be sufficient to prevent different ideas or sentences in the recorded document from running together. This type of rate increase does not affect the intelligibility of individual words within the active speech regions (Maxemchuk 1980).

Studies of pauses in speech also consider the duration of the “non-pause” or “speech unit.” In one study of spontaneous speech, the mean speech unit was 2.3 seconds. Minimum pause durations typically considered in the literature range from 50–800 ms, with the majority in the 250–500 ms region. As the minimum pause duration increases, the mean speech unit length increases (e.g., for pauses of 200, 400, 600, and 800 ms, the corresponding speech unit lengths were 1.15, 1.79, 2.50, and 3.52 s respectively). In another study, it was found that inter-phrase pauses were longer and occurred less frequently than intra-phrase pauses (data from several articles summarized in Agnello 1974).

“Hesitation” pauses are not under the conscious control of the talker, and average 200–250 ms. “Juncture” pauses are under talker control, and average 500–1000 ms. Several studies show that breath stops in oral reading are about 400 ms. In a study of the durational aspects of speech, it was found that the silence and speech unit durations were longer for spontaneous speech than for read speech, and that the overall word rate was slower. The largest changes occurred in the durations of the silence intervals. The greater number of long silence intervals were assumed to reflect the tendency for talkers to hesitate more during spontaneous speech than during oral reading (Minifie 1974). Lass states that juncture pauses are important for comprehension, so they cannot be eliminated or reduced without interfering with comprehension (Lass 1977).

Theories about memory suggest that large-capacity rapid-decay sensory storage is followed by limited capacity perceptual memory. Studies have shown that increasing silence intervals between words increases recall accuracy. Aaronson suggests that for a fixed amount of compression, it may be optimal to delete more from the words than from the intervals between the words (Aaronson 1971). Aaronson states:

> English is so redundant that much of the word can be eliminated without decreasing intelligibility, but the interword intervals are needed for perceptual processing. (Aaronson 1971, 342).

3.8 Summary

This chapter reviewed a variety of techniques for time compressing speech, as well as related perceptual limits of intelligibility and comprehension.

The SOLA method produces the best quality speech for a computationally efficient time domain technique and is currently in vogue for real-time applications. However, a digital version of the Fairbanks sampling method with linear crossfades can easily be implemented, and produces good speech quality with little computation. The sampling technique also lends itself to dichotic presentation for increased comprehension.

For spontaneous or conversational speech the limit of compression is about 50% (2x normal speed). Pauses, at least the short ones, can also be removed from a speech signal, but comprehension may be affected.

4 Adaptive Speech Detection

This chapter presents a survey of the techniques, applications, and problems of automatically discriminating between speech and background noise. An introduction to the basic techniques of speech detection is presented, including a literature survey, and a summary of the techniques in use by the Media Laboratory's Speech Research Group. A variety of analyses of speech recordings are included.

There are two motivations for this work. The primary area of interest is to design an adaptive speech detector to be used with time-compressed speech techniques for pause removal, and for automatically segmenting recordings and finding structure and as part of an exploration of speech skimming (see chapter 5). For example, in Hyperspeech (chapter 2) it was found that the majority of manually selected speech segments began on natural phrase boundaries that coincided with hesitations in the speech recordings (see section 2.2.1). Thus if hesitations can be easily found, it is possible to segment recordings into logical chunks.

The second reason for this work is to investigate techniques for improving the robustness of the Speech Research Group's voice-operated recording system (described in sections 4.3.1 and 4.3.2).

4.1 Introduction

Speech is a non-stationary (time-varying) signal; silence (background noise) is also typically non-stationary. Speech detection[32] involves classifying these two non-stationary signals. "Silence detection" is something of a misnomer since the fundamental problem is in detecting the background noise. Background noise may consist of mechanical noises such as fans, that can be defined temporally and spectrally, but noise can also consist of conversations, movements, and door slams, that are difficult to characterize. Due to the variability of the speech and silence patterns, it is desirable to use an adaptive, or self-normalizing,

[32]The term speech detection is used throughout this document since the speech portions of a signal, rather than the silence, are of primary interest.

solution for discriminating between the two signals that does not rely heavily on arbitrary fixed thresholds (de Souza 1983).

This chapter begins with a detailed description of algorithms used in the Speech Research Group for voice operated recording. Much of the literature found on speech detection under noise conditions, however, is an outgrowth of two research areas: speech recognition and speech interpolation.

In recognizing discrete speech (i.e., isolated words), the end-points of a word must be accurately determined; otherwise recognition algorithms, such as dynamic time warping, may fail. For example, in recognizing the spoken letters of the alphabet (i.e., aye, bee, see, dee, etc.), much of this small vocabulary is distinguished solely by the beginnings and endings of the words—recognition accuracy may be severely reduced by errors in the end-point detection algorithm (Savoji 1989). In end-point detection, however, it is desirable to eliminate speech artifacts such as clicks, pops, lip smacks, and heavy breathing.

“Speech interpolation” is used in the telephone and satellite communication industries for systems that share scarce resources (such as transoceanic channel capacity) by switching telephone conversations during silent intervals. In a telephone conversation, a talker typically speaks for only 40% of the time (Brady 1965); during the silent intervals, the channel is reassigned to another talker. Such a scheme typically doubles the capacity of a bank of telephone lines (Miedema 1962).

Voice activation[33] algorithms, such as those for voice-operated recording or speech interpolation, do not need to be as accurate as for speech recognition systems in determining the start and end points of a signal. Such voice activation schemes, including speech interpolation, usually switch on quickly at low thresholds, and have a “hang-over time” of several hundred milliseconds before turning off, to prevent truncation of words (see section 4.5). In such a system, a small amount of channel capacity or recording efficiency is traded off for conservative speech detection.

4.2 Basic Techniques

Depending on the type of analysis being done, a variety of measures can be used for detecting speech under noise conditions. Five features have

[33]Also called “silence detection” or “pause detection”; sometimes referred to as a “voice operated switch” and abbreviated VOX.

been suggested for voiced/unvoiced/silence classification of speech signals (Atal 1976):

- energy or magnitude
- zero crossing rate (ZCR)[34]
- one sample delay autocorrelation coefficient
- the first LPC predictor coefficient
- LPC prediction error energy

Two or more of these (or similar) parameters are used by most existing speech detection algorithms (Savoji 1989). The computationally expensive parameters are typically used only in systems that have such information readily available. For example, linear prediction coefficients are often used in speech recognition systems that are based on LPC analysis (e.g., Kobatake 1989). Most techniques use at most the first three parameters, of which signal energy or magnitude has been shown to be the best for discriminating speech and silence (see sections 4.4.1 and 4.5.2 for the relative merits of magnitude versus energy). The number of parameters affects the complexity of the algorithm—to achieve good performance, speech detectors that only use one parameter tend to be more complex than those employing multiple metrics (Savoji 1989).

Most of the algorithms use rectangular windows and time-domain measures to calculate the signal metrics as shown in figure 4-1. These measures are typically scaled by 1/N to give an average over the frame; the zero crossing rate is often scaled by F_S/N to normalize the value to zero crossings per second (where F_S is the sampling rate in samples per second, and N is the number of speech samples).

$$\text{magnitude} = \sum_{i=1}^{N} \left| x[i] \right|$$

$$\text{energy} = \sum_{i=1}^{N} \left(x[i] \right)^2$$

$$\text{ZCR} = \sum_{i=1}^{N} \left| \operatorname{sgn}(x[i]) - \operatorname{sgn}(x[i-1]) \right|$$

$$\text{where } \operatorname{sgn}(\text{x[i]}) = \begin{cases} 1 & \text{if } x[i] \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Fig. 4-1. Time-domain speech metrics for frames N samples long.

[34]A high zero-crossing rate indicates low energy fricative sounds such as “s” and “f.” For example, a ZCR greater than 2500 crossings/s indicates the presence of a fricative (O’Shaugnessy 1987; see also section 5.9.3).

The speech detection algorithms make two basic types of errors. The most common is the misclassification of unvoiced consonants or weak voiced segments as background noise. The other type of error occurs at boundaries between speech and silence segments where the classification becomes ambiguous. For example, during weak fricatives the energy typically remains low, making it difficult to separate this signal from background noise. However, the zero crossing rate typically increases during fricatives, and many algorithms combine information from both energy and zero crossing measures to make the speech versus background noise decision. The zero crossing rate during silence is usually comparable with that of voiced speech.

Some algorithms assume that the beginning of the signal is background noise; however for some applications this condition cannot be guaranteed. The requirements for an ideal end-point detector are: reliability, robustness, accuracy, adaptivity, simplicity, and real-timeness without assuming *a priori* knowledge of the background noise (Savoji 1989).

4.3 Pause Detection for Recording

A simple adaptive speech detector based on an energy threshold is used by the Speech Research Group for terminating recordings made over the telephone (i.e., voice mail messages). Applications can set two parameters to adjust the timing characteristics of this voice-operated recorder. The “initial pause time” represents the maximum amount of silence time permitted at the beginning of a recording. Similarly, the “final pause time” is the amount of trailing silence required to stop the recording. For example, an initial pause time of 4 seconds allows talkers a chance to collect their thoughts before starting to speak. If there is no speech during this initial interval, the recording is terminated, and no data is saved. If speech is detected during the initial interval, recording continues until a trailing of silence of the final pause time is encountered.[35] For example, with a final pause time of 2 seconds, there must be two contiguous seconds of silence after speech is detected for recording to stop. The leading and trailing silences are subsequently removed from the data files.

---

[35] Recording will also terminate if a predefined maximum length is reached.

4.3.1 Speech Group Empirical Approach: Schmandt

The speech detection system is used with a Natural Microsystems VBX board running on a Sun 386i workstation. The board records 8-bit 8 kHz µ-law speech from an analog telephone line and provides a “rough log base 2 energy value” every 20 ms (Natural 1988, 22). This value is then used in a simple adaptive energy threshold detector to compensate for differences in background noise across telephone calls and varying quality connections.

The minimum energy value (the “silence threshold”) is tracked throughout a recording. A piecewise linear function maps this value into a “speech threshold” (figure 4-2). Signal values that are below the speech threshold are considered background noise; those above it are considered speech. The mapping function was determined empirically, by manually analyzing the energy patterns from many recordings made by the system under a variety of line, noise, and speaking conditions.

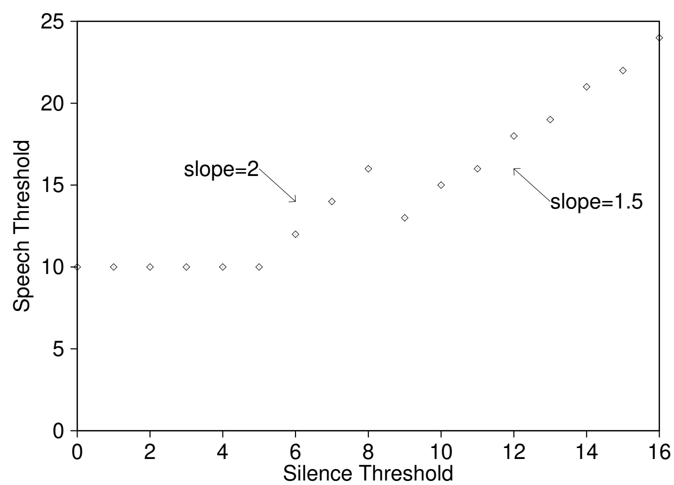


Fig. 4-2. Threshold used in Schmandt algorithm.

If there is only background noise at the beginning of a recording, the silence threshold and speech threshold are set during the first frame, before the caller starts speaking. Because of variations in the background noise, the noise threshold then typically drops by small amounts during the remainder of the recording.

This algorithm is simple and effective as a speech-controlled recording switch, but has several drawbacks:

- The mapping function between the noise threshold and speech threshold must be determined manually. These values are dependent on the recording hardware and the energy metric used, and must be determined for each new hardware configuration supported.

- The algorithm assumes semi-stationary background noise, and may fail if there is an increase in background noise during a recording.
- Since the noise threshold is determined on-the-fly, the algorithm can fail if there is speech during the initial frames of the recording. Under this condition the silence threshold remains at its initial default value, and the algorithm may incorrectly report speech as silence. The default value of the silence threshold, representing the highest level of background noise ever expected, must thus be chosen carefully to minimize this type of error.

4.3.2 Improved Speech Group Algorithm: Arons

In an attempt to implement a pause detection algorithm on new hardware platforms, and to overcome some of the limitations of the Schmandt algorithm (section 4.3.1), a new approach was taken. A pause detection module is called with energy values at regular intervals. This value is then converted to decibels to reduce its dynamic range, and provide a more intuitive measure based on ratios.

The algorithm currently runs in two operating environments:

- On a Sun SparcStation, RMS energy is calculated in real time with a default frame size of 100 ms.
- On Apple Macintoshes, an average magnitude measure is used. The Macintosh Sound Manager is queried during recording to obtain a “meter level” reading indicating the value of the most recent sample. Because of background processing in our application environment, the time between queries ranges from 100 to 350 ms. During a typical interval of 100 ms, the meter is polled roughly seven times, but may be called only once or twice (and is thus similar to the technique described in section 4.3.3).

The complex mapping function used in the Schmandt algorithm is replaced by a simple signal-to-noise (SN) constant. For example, with the SN constant set to 4 dB, if the lowest energy obtained during a recording is 20 dB, the speech threshold is set to 24 dB. Any frames with energy under the speech threshold (i.e., < 24 dB) are judged as background noise, while frames above the threshold (i.e., ≥ 24 dB) are judged as speech (figure 4-3).

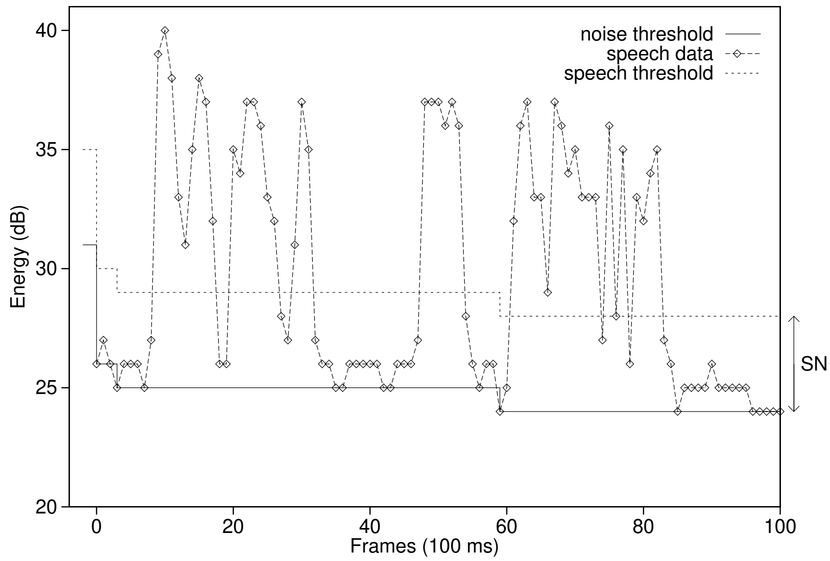


Fig. 4-3. Threshold values for a typical recording. Note that the threshold falls by 6 dB from its default value (31 dB) in the first frame, then decreases by 1 dB two other times.

If there is an initial silence in the signal, the threshold drops to the background noise level during the first frame of the recording. However, if there is speech during the first frame, the threshold is not set to background noise, and a speech segment may be inappropriately judged as silence because it is below the (uninitialized) speech threshold. To overcome this limitation, if there is a significant drop in energy after the first frame, the algorithm behaves as if speech were present since the recording was started (figure 4-4). The value of the drop required currently is set to the same numerical value as the SN constant (i.e., 4 dB).

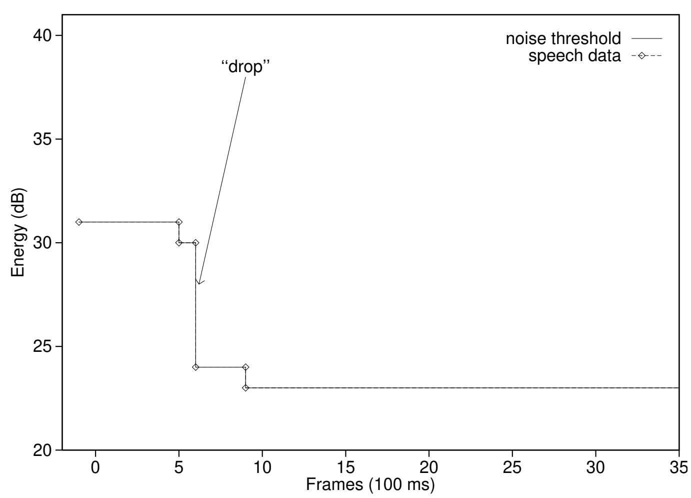


Fig. 4-4. Recording with speech during initial frames. A "drop" in the noise threshold occurs at frame 6, suggesting that speech was present for frames 0–6.

The large window size and simple energy measure used in these two algorithms is crude in comparison to the other techniques described in this chapter, and hence may incorrectly identify weak consonants as silence. For these speech recording applications, however, initial and final pause durations are also typically large (in the 1–4 second range). After a recording has been terminated by pause detection, the leading and trailing silences are truncated. This truncation is done conservatively in case the recording begins or ends with a weak consonant. For example, if the requested trailing pause is 2000 ms, only the last 1900 ms is truncated from the recording.

4.3.3 Fast Energy Calculations: Maxemchuk

Maxemchuk used 62.5 ms frames of speech corresponding to disk blocks (512 bytes of 8 kHz, 8-bit μ-law data). For computational efficiency, only a pseudo-random sample of 32 out of every 512 values were looked at to determine low-energy portions of the signal (Maxemchuk 1980). Several successive frames had to be above or below a threshold in order for a silence or speech determination to be made.

4.3.4 Adding More Speech Metrics: Gan

Gan and Donaldson found that amplitude alone was insufficient to distinguish weak consonants from the background, so a zero crossing metric and two adaptive amplitude thresholds were used to classify each 10 ms frame of a voice mail message (Gan 1988). The algorithm uses four primary parameters:

- the zero crossing threshold between speech and silence
- the minimum continuous amount of time needed for a segment to be classified as speech
- the amplitude threshold for determining a silence-to-speech transition
- the amplitude threshold for determining a speech-to-silence transition

The local average of the ten most recent silence frames determines the background noise. This noise average is multiplied by the amplitude thresholds to adapt to non-stationary noise conditions. The average noise value is initialized to a default value, and all ten values are reset during the first silence frame detected. This technique therefore does not require that the beginning segments of a recording be silence. Short sound bursts, that are inappropriately classified as speech because of energy and ZCR metrics, are eliminated by the minimum speech time requirement.

Note that the four parameters must be tuned for proper operation of the algorithm. Parameters were varied for a 30 second test recording to achieve the highest silence compression without cutting off a predefined set of weak consonants and syllables. The details of the algorithm are straightforward, and the technique was combined and tested with several waveform coding techniques.

4.4 End-point Detection

Rabiner has called locating the end-points essentially a problem of pattern recognition—by eye one would acclimate to a “typical” silence waveform and then try to spot radical changes in the waveform (Rabiner 1975). This approach may not work, however, in utterances that begin or end with weak fricatives, contain weak plosives, or end in nasals.

4.4.1 Early End-pointing: Rabiner

In Rabiner’s algorithm, signal magnitude values are summed as a measure of “energy” (Rabiner 1975). Magnitude is used instead of true energy for two reasons: first, to use integer arithmetic for computational speed and to avoid possible overflow conditions, and second because the magnitude function de-emphasizes large-amplitude speech variations and produces a smoother energy function. The algorithm assumes silence in the first 100 ms, and calculates average energy and zero crossing statistics during that interval. Several thresholds are derived from these measures and are used for end-pointing.

The use of energy is perhaps more physically meaningful than average magnitude, as it gives more weight to sample values that are not near zero. Energy calculations, however, involve a multiplication, and are hence considered more computationally expensive than magnitude computations. Note that on some microprocessors a floating multiply and add are faster than an integer addition.

4.4.2 A Statistical Approach: de Souza

Knowledge of the properties of speech is not required in a purely statistical analysis; it is possible to establish the patterns of the silence, and measure changes in that pattern (de Souza 1983, based on Atal 1976). With a statistical test, arbitrary speech-related thresholds are avoided; only the significance level of the statistical test is required. Setting the significance level to P means that, on average, P percent of the silent frames are mislabeled as speech. A significance level of one percent was found to produce an acceptable tradeoff between the types of errors produced.

The statistical system of de Souza requires that the first 500 ms of the recording be silence to bootstrap the training procedure. The system uses parameters for 10 blocks of silence in a first-in first-out (FIFO) arrangement, discarding the oldest parameters whenever a new 500 ms block of silence is found. This technique allows the system to adapt to a smoothly varying background noise. Training is complete when five seconds of silence data have been processed; the silence detector then returns to the start of the input to begin its final classification of the signal.

Five metrics are computed for each 10 ms frame in the signal. In addition to energy, zero crossings, and the unit sample delay autocorrelation coefficient, two additional metrics attempt to capture the information a person uses when visually analyzing a waveform display. The "jaggedness" is the derivative of the ZCR, and the "shade" is a normalized difference signal measure.[36] The author conceded that the choice of metrics was somewhat arbitrary, but they work well in practice.

4.4.3 Smoothed Histograms: Lamel et al.

Lamel et al. developed an end-point detection algorithm for recordings made over telephone lines (Lamel 1981). The first stage of the algorithm is of most interest in speech detection; the "adaptive level equalizer" normalizes the energy contour to compensate for the mean background noise. This portion of the algorithm is described in further detail in (Wilpon 1984).

The minimum energy is tracked for all frames. Then this background level estimate is refined further by computing a histogram of the energy values within 10–15 dB of the minimum. A three-point averager is applied to the histogram, then the mode (peak) of the histogram is taken as the background noise level.

[36] The base 10 logarithm of two of the parameters was taken to improve the fit of the measure to a normal distribution.

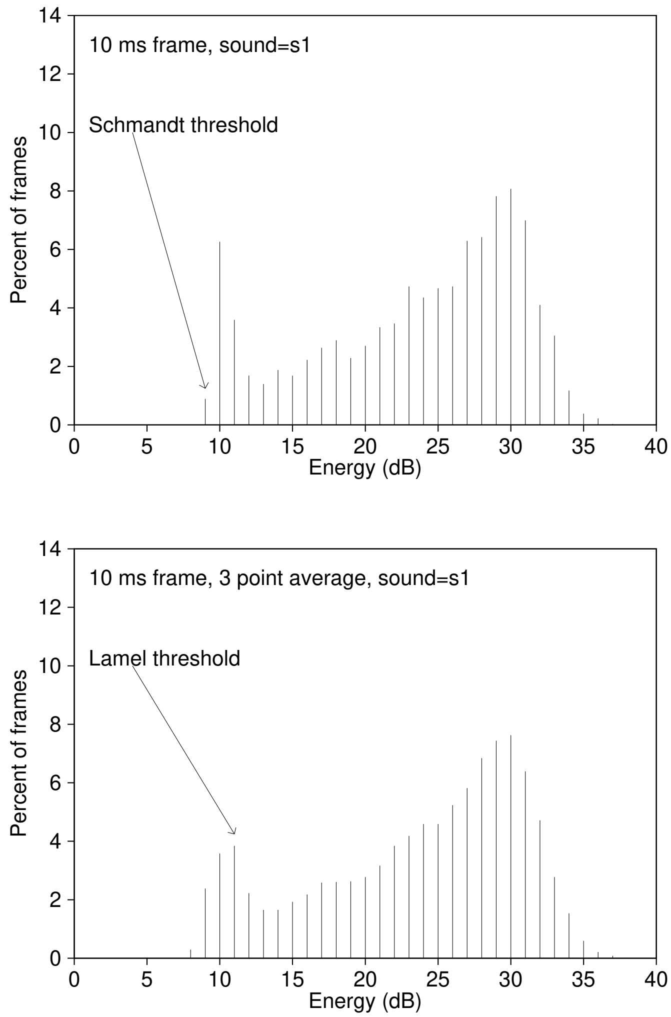


Fig. 4-5. Energy histograms with 10 ms frames. Bottom graph has been smoothed with a three-point filter. Data is 30 seconds of speech recorded over the telephone.

The Lamel noise detector improves on those described in sections 4.3.1 and 4.3.2 by providing a margin of error if the minimum energy value is anomalous, or if the background noise changes slowly over time.[37] Figures 4-5 and 4-6 show energy histograms for speech of a single talker recorded over a telephone connection. Frame sizes of 10 (figure 4-5) and 100 ms (figure 4-6) are shown, including a three-point averaged version of each histogram. The noise thresholds determined by the Lamel and Schmandt algorithms are noted in figure 4-5. Energy values within a constant of the threshold (Lamel, Arons), or determined by a function (Schmandt), are judged as silence. Figure 4-7 shows similar histograms for four different talkers. Note the difference in energy patterns and total

[37]Other algorithms described in this chapter are better at adapting to faster changes in noise level.

percentage of "silence" time for each talker (these data are for telephone monologues).

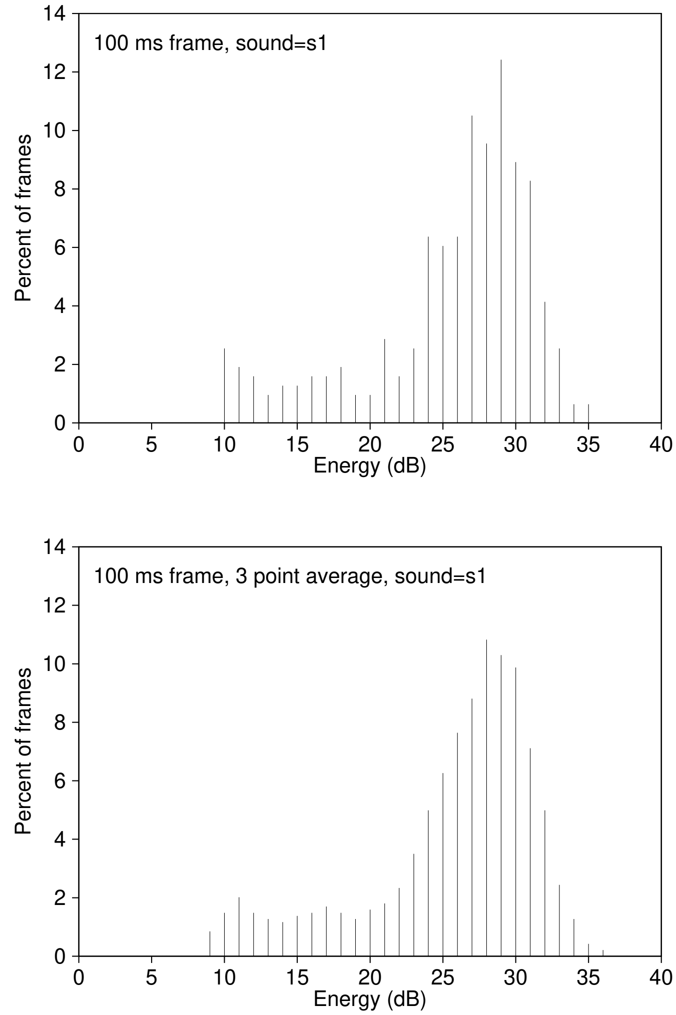


Fig. 4-6. Energy histograms with 100 ms frames. Bottom graph has been smoothed with a three-point filter. The same speech data is used in figure 4-5.

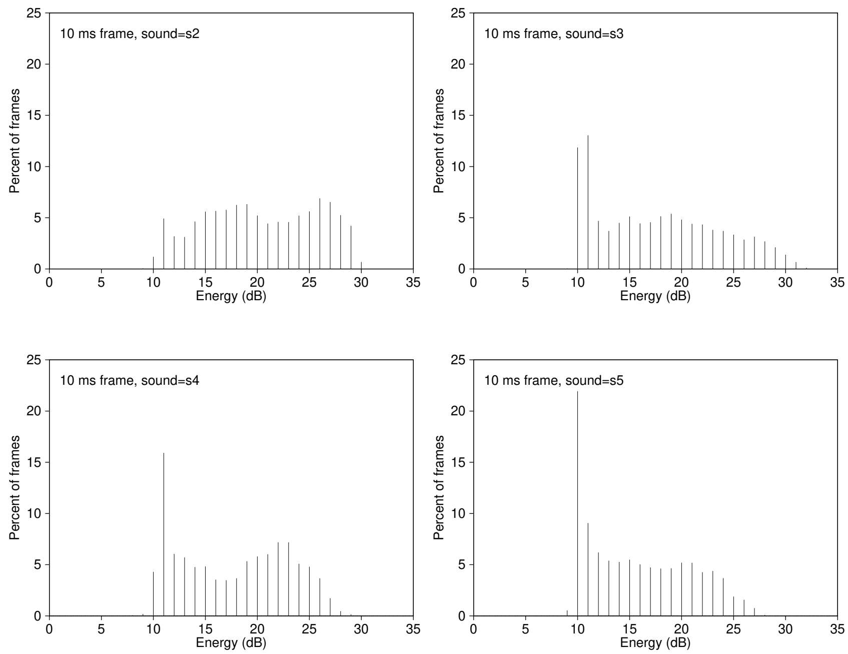


Fig. 4-7. Energy histograms of speech from four different talkers recorded over the telephone.

The end-point detector then uses four energy thresholds and several timing thresholds to determine speech-like bursts of energy in the recording (also summarized in O'Shaughnessy 1987). The application of the thresholds is an attempt to eliminate extraneous noises (e.g., breathing), while retaining low energy phonemes.

Lamel's work is also concerned with implicit versus explicit end-point detection, and develops an improved hybrid approach that combines end-pointing with the speech recognition algorithm. Wilpon said that Lamel's bottom-up algorithm works well in stationary noise with high signal-to-noise ratios, but it fails under conditions of variable noise (Wilpon 1984). Wilpon retained the same adaptive level equalizer, but improved on the performance of the end-pointing algorithm by using top-down syntactic and semantic information from the recognition algorithm (Wilpon 1984).

4.4.4 Signal Difference Histograms: Hess

Hess recognized silences by capitalizing on the fact that histograms[38] of energy levels tend to have peaks at levels corresponding to silence (Hess 1976). It was noted that since the speech signal level changes much faster than the semi-stationary background noise, a histogram shows a distinct maximum at the noise level. A threshold above this peak is then derived

[38]This section is included in the end-pointing portion of this chapter because this earlier paper ties in closely with the histograms used by Lamel et al. (section 4.4.3).

for separating speech from silence. To adapt to varying levels of background noise, the entire histogram was multiplied by a constant (less than 1.0) when one value in the histogram exceeded a predefined threshold.

To help identify weak fricatives (which may be confused with noise), a histogram was also made of the magnitude of the differenced signal:

$$\text{differenced magnitude} = \sum_{i=1}^{N} \left| x[i] - x[i-1] \right|$$

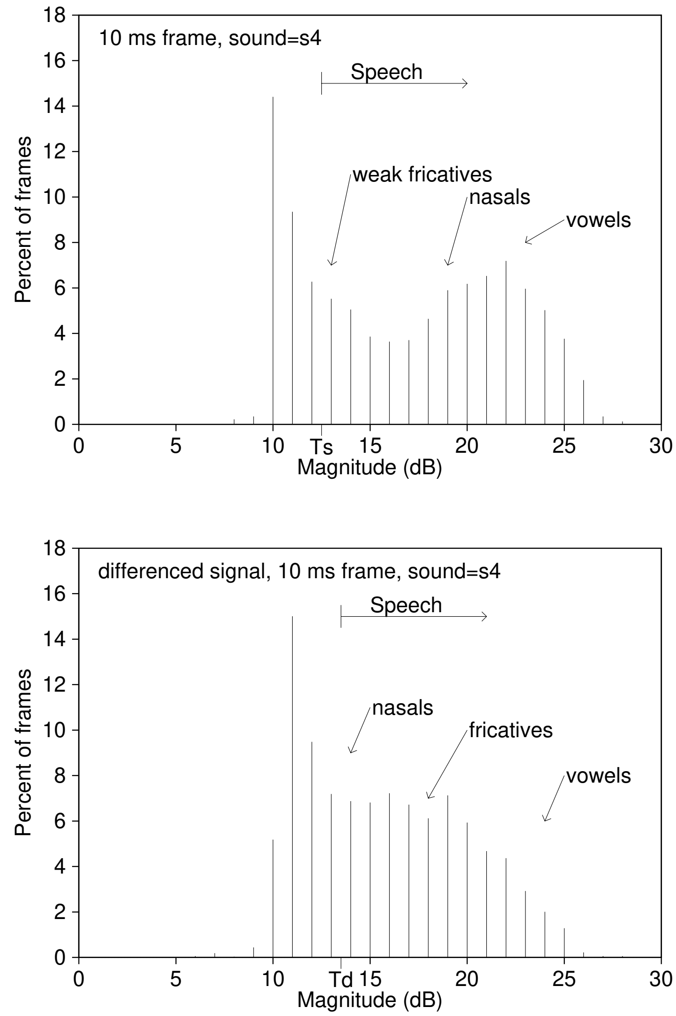


Fig. 4-8. Signal and differenced signal magnitude histograms. Note that the speech thresholds for the signal (Ts) and the differenced signal (Td) are a fixed amount above the noise threshold. The phonetic categorizations are after (Hess 1976).

In figure 4-8, Ts is the speech threshold for the signal, and is set above the noise threshold for the signal. Td is similarly the speech threshold for the differenced signal. Each frame has two metrics associated with it; the magnitude level of the signal (Ls), and the magnitude level of the differenced signal (Ld). A frame is classified as silence if (Ls < Ts) and (Ld < Td); otherwise it is speech.

4.4.5 Conversational Speech Production Rules: Lynch et al.

Lynch et al. present a technique for separating speech from silence segments, and an efficient method of encoding the silence for later reconstruction as part of a speech compression system (Lynch 1987). The algorithm uses a few simple production rules and draws on statistical analyses of conversational telephone speech (Brady 1965; Brady 1968; Brady 1969; see also Lee 1986). For example, the empirical evidence shows that 99.9% of continuous speech spurts last less then 2.0 seconds, and that such speech contains short (<150 ms) intersyllabic gaps. The production rules based on these data allow the background noise level to be tracked in real time. If there is non-stationary noise, the system adapts instantly if the noise level is decreased. If the noise level is increased, there is a lag of about 5 seconds before the system adapts because of the time constants used in the production rules.

Removing silences in this manner has little effect on perceived quality if the signal-to-noise ratio (SNR) is at least 20 dB. Quality is degraded if the SNR is between 10–20 dB because of the clipping of low-level sounds at the ends of speech segments. Below 10 dB SNR, intelligibility is degraded from misclassifications of speech as noise. Lynch et al. report that the silence reconstruction[39] does not affect intelligibility.

This technique was subsequently modified, including the addition of zero crossing rate detection, to create a robust end-point detector (Savoji 1989).

4.5 Speech Interpolation Systems

TASI (Time Assigned Speech Interpolation) was used to approximately double the capacity of existing transoceanic telephone cables (Miedema 1962). Talkers were assigned to a specific channel while they were speaking; the channel was then freed during silence intervals. During busy hours, a talker was assigned to a different channel about every other "talkspurt." The TASI speech detector was necessarily a real time device, and was designed to be sensitive enough to prevent clipping of the first syllable. However, if it is too sensitive, the detector triggers on noise and

[39]The silence reconstruction is based on an 18th-order polynomial with only three non zero terms. This produces a pseudo-random noise sequence with a long (33 s) repetition rate.

the system operates inefficiently. The turn-on time for the TASI speech detector is 5 ms, while the release time is 240 ms. The newer DSI (Digital Speech Interpolation) technique is similar, but works entirely in the digital domain.

If the capacity of a speech interpolation system is exceeded, a conversation occupying the transmission channel will "freeze out" other conversations that attempt to occupy the channel (Campanella 1976).[40] A DSI system is more flexible, allowing the quality of several channels to be slightly degraded for a short time, rather than completely freezing out conversation.[41] Such changes are not apparent to the conversants.

"Hangover" bridges short silences in speech, and creates fewer, but longer talkspurts, thus reducing the effects of variable network delays. Hangover times ≥ 150 ms are recommended, with 200 ms as a typical value (Gruber 1983). An alternative to the hangover technique, called "fill-in," eliminates silences shorter than the fill-in time (Gruber 1982). A delay equal to the fill-in time is required (often 200 ms), suggesting that the technique be used for non real-time applications such as voice response systems. The fill-in technique produces higher speech activity[42] than the hangover technique, producing longer average silences and shorter average talkspurts (figure 4-9).

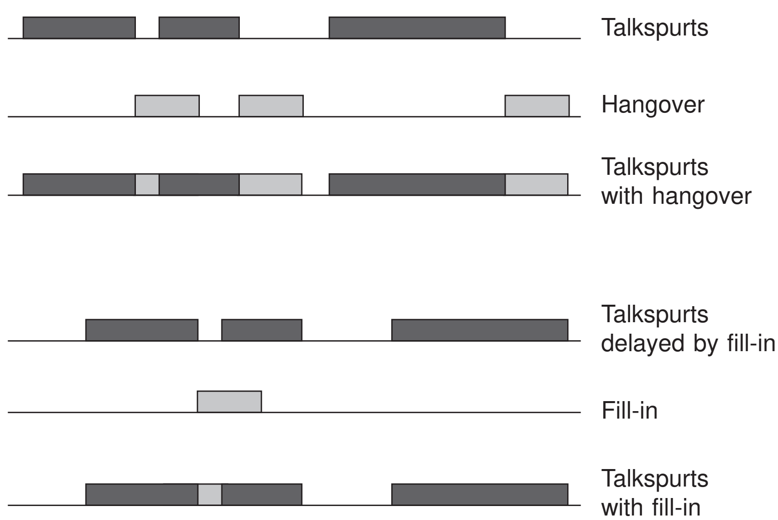


Fig. 4-9. Hangover and fill-in (after Gruber 1983).

The loss of the initial portion of a speech spurt is called front-end clipping (FEC). A FEC duration of 15 ms is approximately the threshold

[40]The freeze-out fraction is typically designed to be less than 0.5 percent.
[41]The standard technique is to allocate 7 quantizing bits to the channel instead of the normal 8, adding 6 dB of quantization noise.
[42]Speech activity is the ratio of talkspurt time to total time.

of perceptibility (Gruber 1983).[43] FECs of less than 50 ms provide good quality, but clipping of > 50 ms potentially affects intelligibility.

4.5.1 Short-term Energy Variations: Yatsuzuka

A sensitive speech detector based on energy, zero crossing rates, and sign bit sequences in the input signal was developed for a DSI environment (Yatsuzuka 1982). The speech detector is defined by a finite state machine with states representing speech, silence, hangover, and the “primary detection” of speech before speech is fully recognized. In addition to the absolute level of energy, the short-term variation of energy between adjacent 4 ms frames assists in detecting the silence-to-speech transition. A periodicity test on the sign bit sequences of the signal is used when it is difficult to discriminate between speech and silence.

4.5.2 Use of Speech Envelope: Drago et al.

Speech exhibits great variability in short-time energy, while the background noise on telephone channels is semi-stationary and has only slightly variable short-time energy. Good speech detection results have been obtained by analyzing the short-time energy of the speech channel (Drago 1978). Magnitude, rather than energy, was used for simplicity and because the squaring operation reduces the relative effect of small amplitude signals. This suggests that energy is a better measure than magnitude as it makes a larger distinction between speech and silence. The dynamic speech detector relied on the relative variation in the envelope of the signal. Noise is considered as a random process with small short-time variations in the envelope, while speech has a highly variable envelope.

4.5.3 Fast Trigger and Gaussian Noise: Jankowski

Design criteria for a new voice-activated switch for a satellite-based DSI system included fast threshold adjustment to variable noise, improved immunity to false detection of noise, and no noticeable clipping of speech (Jankowski 1976). Three thresholds are used in the system:

1. the noise threshold;
2. the speech threshold (7 quantizing steps above the noise level);
3. a threshold that disables noise adaptation during speech.

---

[43] It is recommended that the total amount of speech loss be limited to ≤ 0.5%.

Only three samples of speech (375 µs) above the first threshold are needed to trigger the voice switch. The observation window was kept as short as possible to minimize the front end clipping of a talkspurt. A delay of 4 ms is inserted in the signal path, so that speech is effectively turned on 3.625 ms before the switch triggers. Once speech is detected, there is a 170 ms hangover time.

Telephone noise can be considered as a Gaussian distribution, and a noise threshold was taken as the 96th percentile point of the noise distribution. To establish a 10% error criterion for this measurement, 1200 samples (150 ms at 8 kHz sampling) of speech are required to determine the noise threshold. The noise threshold is adjusted during 150 ms periods of silence so that 4% of the noise samples are above the threshold. If more than 5% of the samples (60 samples) are above the threshold, the threshold is raised by one quantizing step. If less than 3.3% of the samples (40 samples) are below the threshold, it is reduced by one step.

4.6 Adapting to the User's Speaking Style

An earlier version of the Schmandt technique was used in the Phone Slave conversational answering machine (see section 1.4.3). In addition to adapting to the background noise, the length of the final pause was also determined adaptively (Schmandt 1984). The final pause was initialized to 1.25 seconds, but if there were intermediate pauses greater than 750 ms, the final pause length was gradually increased up to a maximum of 2 seconds. This adaptive pause length detector prevented slow talkers who pause a lot from being cut off too soon, yet permitted fast response for rapid or terse talkers. This type of adaptation was important to enable the conversational interaction style developed in Phone Slave.

Using parameters similar to those used in TASI/DSI systems, Watanabe investigated adapting the speech rate[44] of a conversational answering machine with the speech rate of the user (Watanabe 1990). The speech activity of the user was found to correlate strongly with the speech speed—talkers with higher speech activity ratios speak faster. This metric was used to set the speech activity of a synthesizer to match the on-off pattern of the talker to realize a smooth information exchange between the human and machine.

[44] Measured in syllable-like units per second.

4.7 Summary

This chapter reviewed literature on detecting speech versus background noise, focusing on simple techniques that are adaptive and do not require particular recording characteristics (such as silence at the beginning of a recording) or manually set thresholds. Two algorithms used within the Speech Research Group are described, including an improved technique that can be used to terminate speech recordings under a variety of noise conditions. This chapter also presents a variety of histograms used as analysis tools for understanding conversational speech and developing an appropriate speech detector to be used to automatically segment speech recordings (see section 5.9.3). Note that some of the techniques presented in this chapter must run in real time (e.g., speech interpolation), but for some speech applications, such as skimming recordings, it is feasible to analyze the whole recording to adapt the speech detection parameters to the recorded data.

5 SpeechSkimmer

> *He would have no opportunity to re-listen, to add redundancy by repetition, as he can by re-reading visual displays.… the listener should be given some control over the output pacing of auditory displays. A recommended design solution is to break up the computer output into spoken sentences or paragraphs so that user interaction with the system becomes a transactional sequence.*
>
> (Smith 1970, 219)

Previous chapters have outlined the difficulties of skimming recorded speech, and described some fundamental technologies that can be applied to the problem. This chapter integrates these and new ideas into a coherent system for interactive listening.[45] A framework is described for presenting a continuum of time compression and skimming techniques. For example, this allows a user to quickly skim a speech message to find portions of interest, then use time compression for efficient browsing of the recorded information, and then slow down further to listen to detailed information.

By exploiting properties of spontaneous speech it is possible to automatically select and present salient audio segments in a time-efficient manner. This chapter describes pause- and pitch-based techniques for segmenting recordings and an experimental user interface for skimming speech. The system incorporates time-compressed speech and pause removal to reduce the time needed to listen to speech recordings. This chapter presents a multi-level approach to auditory skimming, along with user interface techniques for interacting with the audio and providing feedback. The results of a usability test are also discussed.

5.1 Introduction

This chapter describes *SpeechSkimmer*, a user interface for skimming speech recordings. SpeechSkimmer uses simple speech processing

[45]Portions of this chapter originally appeared in Arons 1993a.

techniques to allow a user to hear recorded sounds quickly, and at several levels of detail. User interaction through a manual input device provides continuous real-time control over the speed and detail level of the audio presentation.

SpeechSkimmer explores a new paradigm for interactively skimming and retrieving information in speech interfaces. This research takes advantage of knowledge of the speech communication process by exploiting structure, features, and redundancies inherent in spontaneous speech. Talkers embed lexical, syntactic, semantic and turn-taking information into their speech as they have conversations and articulate their ideas (Levelt 1989). These cues are realized in the speech signal, often as hesitations or changes in pitch and energy.

Speech also contains redundant information; high-level syntactic and semantic constraints of English allow us to understand speech when it is severely degraded by noise, or even if entire words or phrases are removed. Within words there are other redundancies that allow partial or entire phonemes to be removed while still retaining intelligibility.

This research attempts to exploit acoustic cues to segment recorded speech into semantically meaningful chunks. The recordings are then time-compressed to further remove redundant speech information. While there are practical limits to time compression, there are compelling reasons to be able to quickly skim a large speech document. For skimming, redundant as well as non-redundant segments of speech must be removed. Ideally, as the skimming speed increases, the segments with the least information content are eliminated first.

When searching for information visually, we tend to refine our search over time, looking successively at more detail. For example, we may glance at a shelf of books to select an appropriate title, flip through the pages to find a relevant chapter, skim headings to find the right section, then alternately skim and read the text until we find the desired information. To skim and browse recorded speech in an analogous manner the listener must have interactive control over the level of detail, rate of playback, and style of presentation. *SpeechSkimmer allows a user to control the auditory presentation through a simple interaction mechanism that changes the granularity, time scale, and style of presentation of the recording.*

This research introduces a new way to think about skimming and finding information in speech interfaces by combining information from multiple sources into a system that allows interactive retrieval (figure 5-1). Skimming, as defined in section 1.1, means automatically selecting and presenting short segments of speech under the user's control. Note that this form of machine-mediated supervisory control (Sheridan 1992a; Sheridan 1992b) is significantly different from skimming a scene with the eyes.

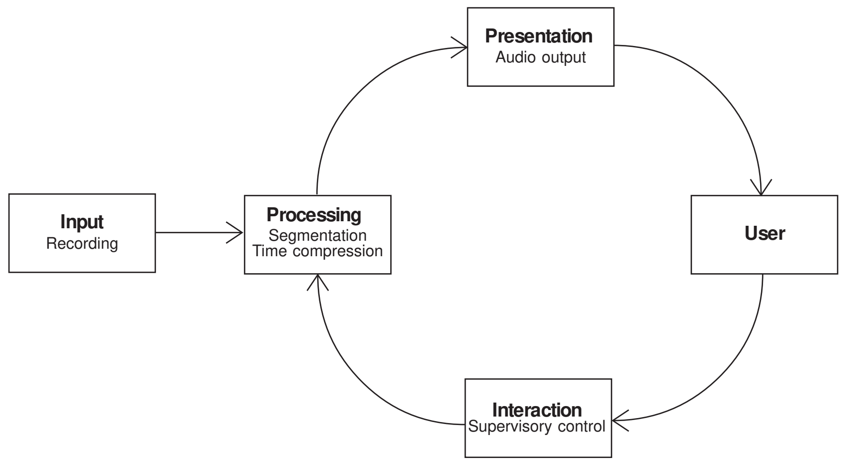


Fig. 5-1. Block diagram of the interaction cycle of the speech skimming system.

5.2 Time compression and Skimming

A variety of speech time compression techniques have been investigated during the background research for this dissertation (see chapter 3). This new research incorporates ideas and techniques from conventional time compression algorithms, and attempts to go beyond the 2x perceptual barrier typically associated with time compressing speech. These new skimming techniques are intimately tied to user interaction to provide a range of audio presentation speeds. Backward variants of the techniques are also developed to allow audio recordings to be played and skimmed backward as well as forwards. The range of speeds and corresponding levels of abstraction are shown in figures 5-2 and 5-3.

1. Normal
2. Time-compressed
   - Silence removal
   - Sampling
   - SOLA
   - Dichotic sampling
   - Combined time compression techniques
   - Backward sampling (for intelligible rewind)
3. Skimming
   - Isochronous skimming (equal time intervals)
   - Speech synchronous skimming (pause- or pitch-based)
   - Backward skimming

Fig. 5-2. Ranges and techniques of time compression and skimming.

Time compression can be considered as "content lossless" since the goal is to present all the non-redundant speech information in the signal. The skimming techniques are designed to be "content lossy," as large parts of the speech signal are explicitly removed. This classification is not based on the traditional engineering concept of lossy versus lossless, but is based on the intent of the processing. For example, isochronous skimming selects and presents speech segments based on equal time intervals. Only the first five seconds of each minute of speech may be played; this can be considered coarse and lossy sampling. In contrast, a speech synchronous technique selects important or emphasized words and phrases based on natural boundaries in the speech so that less content is lost.

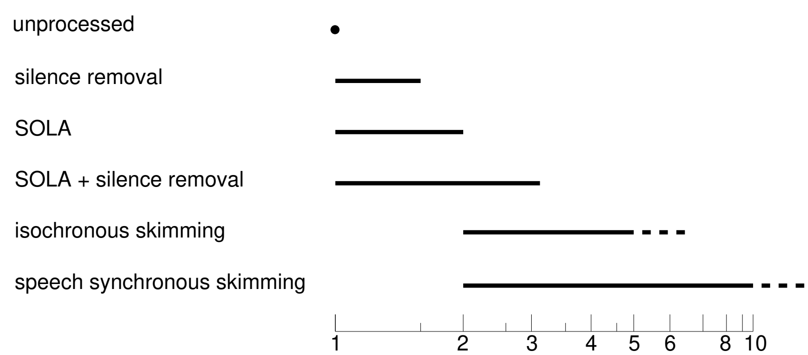


Fig. 5-3. Schematic representation of time compression and skimming ranges. The horizontal axis is the speed increase factor.

5.3 Skimming Levels

There have been a variety of attempts to present hierarchical or "fisheye" views of visual information (Furnas 1986; Mackinlay 1991). These approaches are powerful but inherently rely on a spatial organization. Temporal video information has been displayed in a similar form (Mills 1992), yet this primarily consists of mapping time-varying spatial information into the spatial domain. Graphical techniques can be used for a waveform or similar display of an audio signal, but such a representation is inappropriate—*sounds need to be heard, not viewed.* This research attempts to present a hierarchical (or "fish ear") representation of audio information that *only* exists temporally.

A continuum of time compression and skimming techniques have been designed, allowing a user to efficiently skim a speech recording to find portions of interest, then listen to it time-compressed to allow quick browsing of the recorded information, and then slow down further to listen to detailed information. Figure 5-4 presents one possible "fish ear" view of this continuum. For example, what may take 60 seconds to listen to at normal speed may take 30 seconds when time-compressed, and only five or ten seconds at successively higher levels of skimming. If the speech segments are chosen appropriately, it is hypothesized that this mechanism provides a summarizing view of a speech recording.

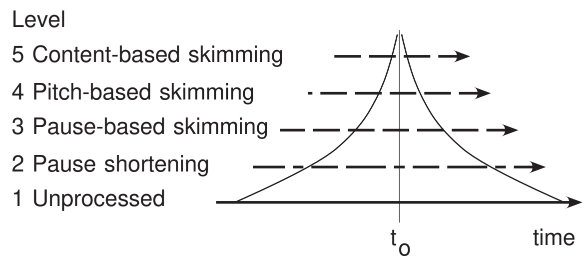


Fig. 5-4. The hierarchical "fish ear" time-scale continuum. Each level in the diagram represents successively larger portions of the levels below it. The curves represent iso-content lines, i.e., an equivalent time mapping from one level to the next. The current location in the sound file is represented by t_o; the speed and direction of movement of this point depend upon the skimming level.

Four distinct skimming levels have been implemented (figure 5-5). Within each level the speech signal can also be time-compressed. The lowest skimming level (level 1) consists of the original speech recording without any processing, and thus maintains the pace and timing of the original signal. In level 2 skimming, the pauses are selectively shortened or removed. Pauses less than 500 ms are removed, and the remaining pauses are shortened to 500 ms. This technique speeds up listening yet provides the listener with cognitive processing time and cues to the important juncture pauses.

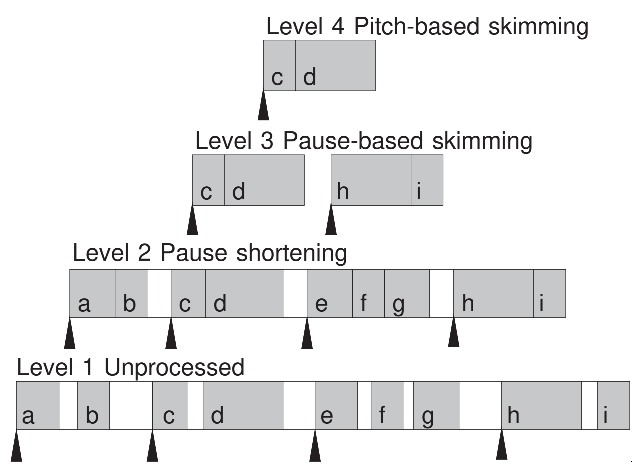


Fig. 5-5. Speech and silence segments played at each skimming level. The gray boxes represent speech; white boxes represent background noise. The pointers indicate valid segments to go to when jumping or playing backward.

Level 3 is based on the premise that long juncture pauses tend to indicate either a new topic, some content words, or a new talker. For example, filled pauses (i.e., "uhh") usually indicate that the talker does not want to be interrupted, while long unfilled pauses (i.e., silences) act as a cue to the listener to begin speaking (Levelt 1989; O'Shaughnessy 1992). Thus level 3 skimming attempts to play salient segments based on this simple heuristic. Only the speech that occurs just after a significant pause in the original recording is played. For example, after detecting a pause over 750 ms, the subsequent 5 seconds of speech are played (with pauses removed). Note that this segmentation process is error prone, but these errors are partially overcome by giving the user interactive control of the presentation. Sections 5.9.3 and 5.9.4 describe the speech detection and segmentation algorithms.

Level 4 is similar to level 3 in that it attempts to present segments of speech that are highlights of the recording. Level 4 segments are chosen by analyzing the pitch, or intonation, of the recorded speech. For example, when a talker introduces a new topic there tends to be an associated increase in pitch range (Hirschberg 1992; Hirschberg 1986; Silverman 1987).[46] Section 5.9.5 details the pitch-based segmentation algorithm. In practice, either level 3 or level 4 is used as the top skimming level.

It is somewhat difficult to listen to level 3 or level 4 skimmed speech, as relatively short unconnected segments are played in rapid succession. It has been informally found that slowing down the speech is useful when skimming unfamiliar material. In this skimming level, a short (600 ms) pure silence is inserted between each of the speech segments. An earlier version played several hundred milliseconds of the recorded ambient noise between segments, but this fit in so naturally with the speech that it was difficult to distinguish between segments.

5.3.1 Skimming Backward

Paul is dead.

Reportedly heard when playing the Beatles' Abbey Road album backward.

Besides skimming forward through a recording, it is desirable to play intelligible speech while interactively searching or "rewinding" through a digital audio file (Arons 1991b; Elliott 1993). Analog tape systems provide little useful information about the signal when it is played completely backward.[47] Silences or other non-speech sounds (such as

[46]Note that "pitch range" is often used to mean the range above the talker's baseline pitch (i.e., the talker's lowest F0 for all speech).
[47]This is analogous to taking "this is a test" and presenting it as "tset a is siht."

beeps or tones) can be easily detected by a listener, and talkers can even be identified since the spectrum is unchanged, but words remain unintelligible.

Digital systems allow word- or phrase-sized chunks of speech to be played forward individually, with the segments themselves presented in reverse order.[48] While the general sense of the recording is reversed and jumbled, each segment is identifiable and intelligible. It can thus become practical to browse backward through a recording to find a particular word or phrase. This method is particularly effective if the segment boundaries are chosen to correspond to periods of silence. Note that this technique can also be combined with time-compressed playback, allowing both backward and forward skimming at high speeds.

In addition to the forward skimming levels, the recorded sounds can also be skimmed backward. Small segments of sound are each played normally, but are presented in reverse order. When level 3 skimming is played backward (considered level –3) the selected segments are played in reverse order. In figure 5-5, skimming level –3 plays segments h–i, then segments c–d. When level 1 and level 2 sounds are played backward (i.e., level –1 and level –2), short segments are selected and played based upon speech detection. In figure 5-5 level –1 would play segments in this order: h–i, e–f–g, c–d, a–b. Level –2 is similar, but without the pauses.

5.4 Jumping

Besides controlling the skimming and time compression, it is desirable to be able to interactively jump between segments within each skimming level. When the user has determined that the segment being played is not of interest, it is possible to go on to the next segment without being forced to listen to each entire segment (see chapter 2 and Resnick 1992a). For example, in figure 5-5 at level 3, segments c and d would be played, then a short silence, then segments h and i. At any time while the user is listening to segment c or d, a jump forward command would immediately interrupt the current audio output and start playing segment h. While listening to segment h or i, the user could jump backward, causing segment c to be played. Valid segments for jumping are indicated with pointers in figure 5-5.

Recent iterations of the skimming user interface have included a control that jumps backward one segment and drops into normal play mode (level 1, no time compression). The intent of this control is to encourage

[48]This method, for example, could result in a presentation of "test, is a, this."

high-speed browsing of time-compressed level 3 or level 4 speech. When the user hears something of interest, it is easy to back up a bit and hear the piece of interest at normal speed.

5.5 Interaction Mappings

A variety of interaction devices (i.e., mouse, trackball, joystick, and touchpad) have been experimented with in SpeechSkimmer. Finding an appropriate mapping between the input devices and controls for interacting with the skimmed speech has been difficult, as there are many independent variables that can be controlled. For this prototype, the primary variables of interest are time compression and skimming level, with all others (e.g., pause removal parameters and pause-based skimming timing parameters) held constant.

Several mappings of user input to time compression and skimming level have been tried. A two-dimensional controller (e.g., a mouse) allows two variables to be changed independently. For example, the y-axis is used to control the amount of time compression while the x-axis controls the skimming level (figure 5-6). Movement toward the top increases time compression; movement toward the right increases the skimming level. The right half is used for skimming forward, the left half for skimming backward. Moving to the upper right thus presents skimmed speech at high speed.

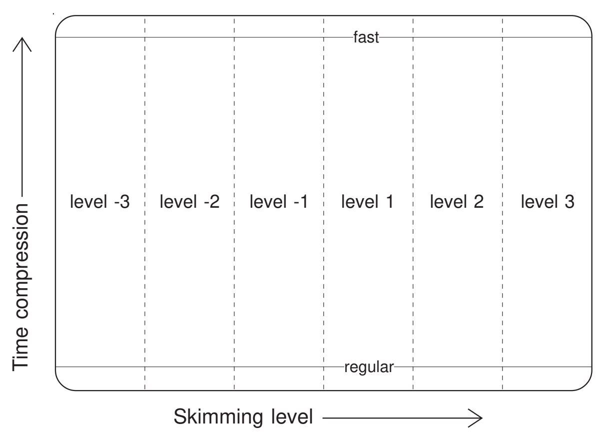


Fig. 5-6. Schematic representation of two-dimensional control regions. Vertical movement changes the time compression; horizontal movement changes the skimming level.

The two primary variables can also be set by a one-dimensional control. For example, as the controller is moved forward, the sound playback speed is increased using time compression. As it is pushed forward further, time compression increases until a boundary into the next level of skimming is crossed. Pushing forward within each skimming level similarly increases the time compression (figure 5-7). Pulling backward has an analogous but reverse effect. Note that using such a scheme with a 2-D controller leaves the other dimension available for setting other parameters.

One consideration in all these schemes is the continuity of speeds when transitioning from one skimming level to the next. In figure 5-7, for example, when moving from fast level 2 skimmed speech to level 3 speech there is a sudden change in speed at the border between the two skimming levels. Depending upon the details of the implementation, fast level 2 speech may be effectively faster or slower than regular level 3 speech. This problem also exists with a 2-D control scheme—to increase effective playback speed currently requires a zigzag motion through skimming and time compression levels.

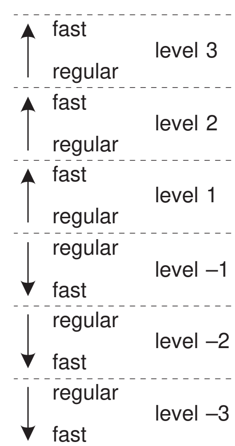


Fig. 5-7. Schematic representation of one-dimensional control regions.

5.6 Interaction Devices

The speech skimming software has been used with a mouse, small trackball, touchpad, and a joystick in both the one- and two-dimensional control configurations.

A mouse provides accurate control, but as a relative pointing device (Card 1991) it is difficult to use without a display. A small hand-held trackball (controlled with the thumb, see figure 5-8) eliminates the desk space required by the mouse, but is still a relative device and is also inappropriate for a non-visual task.

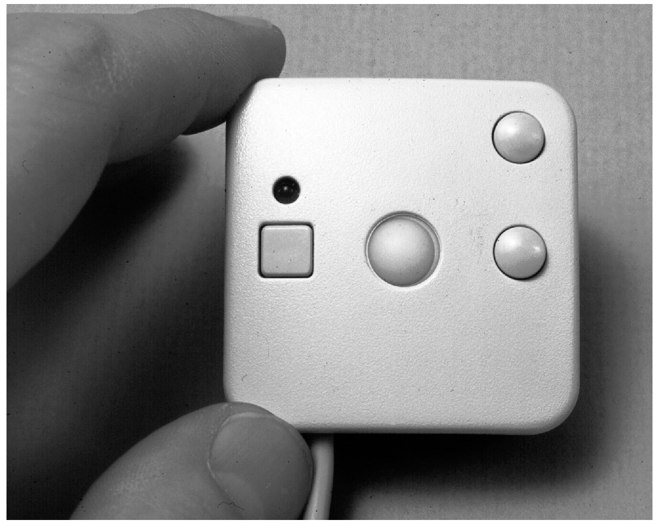


Fig. 5-8. Photograph of the thumb-operated trackball tested with SpeechSkimmer.

A joystick (figure 5-9) can be used as an absolute position device. However, if it is spring-loaded (i.e., automatic return to center), it requires constant physical attention to hold it in position. If the springs are disabled, a particular position (i.e., time compression and skimming level) can be automatically maintained when the hand is removed (see Lipscomb 1993 for a discussion of such physical considerations). The home (center) position, for example, can be configured to play forward (level 1) at normal speed. Touching or looking at the joystick's position provides feedback to the current settings. However, in either configuration, an off-the-shelf joystick does not provide any physical feedback when the user is changing from one discrete skimming level to another, and it is difficult to jump to an absolute location.

Fig. 5-9. Photograph of the joystick tested with SpeechSkimmer.

A small touchpad can act as an absolute pointing device and does not require any effort to maintain the last position selected. A touchpad can be easily modified to provide a physical indication of the boundaries between skimming levels. Unfortunately, a touchpad does not provide any physical indication of the current location once the finger is removed from the surface.

5.7 Touchpad Configuration

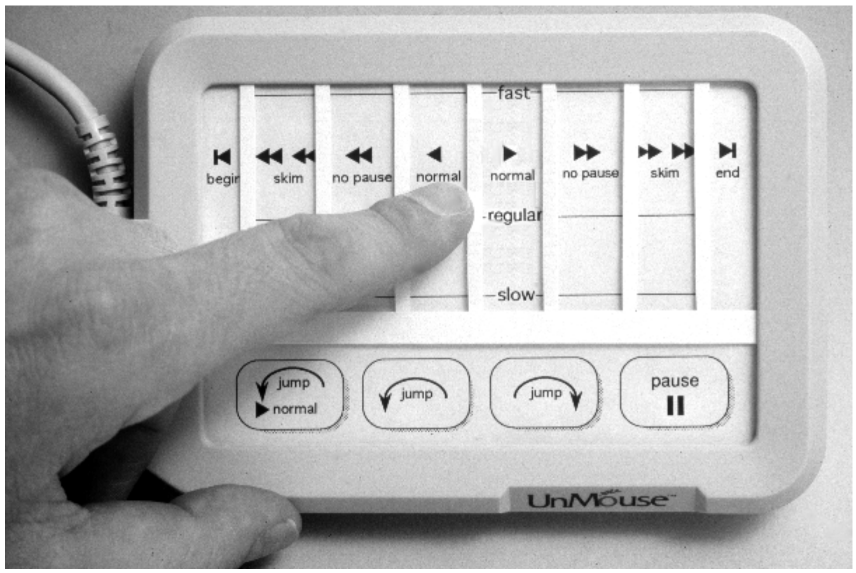


Fig. 5-10. The touchpad with paper guides for tactile feedback.

Currently, the preferred interaction device is a small (7 x 11 cm) touchpad (Microtouch 1992) with the two-dimensional control scheme. This provides independent control of the playback speed and skimming level. Thin strips of paper have been added to the touch-sensitive surface as tactile guides to indicate the boundaries between skimming regions (figure 5-10).[49]

In addition to the six regions representing the different skimming levels,[50] two additional regions were added to enable the user to go to the beginning and end of the sound file. Four buttons provide jumping and pausing capabilities (figure 5-11).

[49]The ability to push down on the surface of the touchpad (to cause a mouse click) has also been mechanically disabled.
[50]As noted in section 5.3, either level 3 or level 4 is used as the top skimming level.

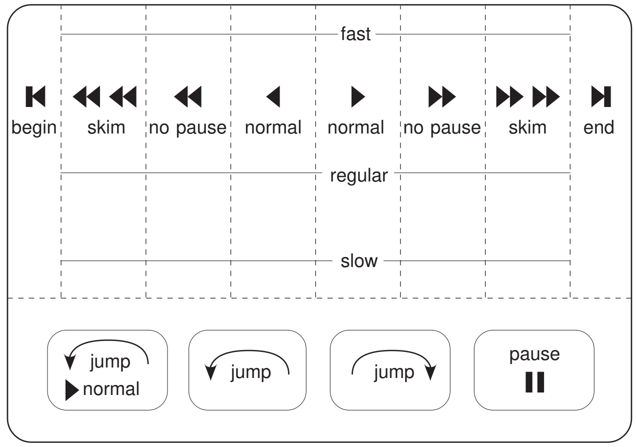


Fig. 5-11. Template used in the touchpad. The dashed lines indicate the location of the guide strips.

The time compression control (vertical motion) is not continuous, but provides a "finger-sized" region around the "regular" mark that plays at normal speed (figure 5-12). To enable fine-grained control of the time compression (Stifelman 1992b), a larger region is allocated for speeding the speech up than for slowing it down. The areas between the tactile guides form virtual sliders (as in a graphical equalizer) that control the time compression within a skimming level.[51]

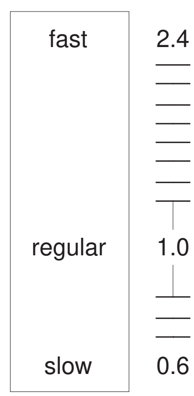


Fig. 5-12. Mapping of the touchpad control to the time compression range.

5.8 Non-Speech Audio Feedback

Since SpeechSkimmer is intended to be used without a visual display, recorded sound effects are used to provide feedback when navigating in the interface (Buxton 1991; Gaver 1989a). Non-speech audio was selected to provide terse, yet unobtrusive navigational cues (Stifelman

[51]In graphical equalizers all the controls are active at once. In this system only one slider is active at a time.

1993).[52] For example, when the user plays past the end or beginning of a sound, a cartoon "boing" is played.

When the user transitions to a new skimming level, a short tone is played. The frequency of the tone increases with the skimming level (i.e., level 1 is 400 Hz, level 2 is 600 Hz, etc.). A double beep is played when the user changes to normal (level 1). This acts as an audio landmark, clearly distinguishing it from the other tones and skimming levels.

No explicit feedback is provided for changes in time compression. The speed changes occur with low latency and are readily apparent in the speech signal itself.

5.9 Acoustically Based Segmentation

Annotating speech or audio recordings by hand can produce high-quality segmentation, but it is difficult, time consuming, and expensive. Automatically segmenting the audio and finding its inherent structure (Hawley 1993) is essential for the success of future speech-based systems. "Finding the structure" here is used to mean finding important or emphasized portions of a recording, and locating the equivalent of paragraphs or new topic boundaries for the sake of creating audio overviews or outlines.

Speech recordings need to be segmented into manageable pieces before presentation. Ideally, a hierarchy of perceptually salient segments can be created that roughly correspond to the spoken equivalents of sentences, paragraphs, and sections of a written document.

Two non-lexical acoustic cues have been explored for segmenting speech:

- *Pauses* can suggest the beginning of a new sentence, thought, or topic. Studies have shown that pause lengths are correlated with the type of pause and its importance (see section 3.7.4).
- *Pitch* is similarly correlated with a talker's emphasis and new topic introductions.

Note that none of these techniques are 100% accurate at finding important boundaries in speech recordings—they all produce incorrect rejections and false acceptances. While it is important to minimize these errors, *it is perhaps more important to be able to handle errors when they occur, as no such recognition technology will ever be perfect.* This

[52]The amount of feedback is user configurable.

research addresses the issues of using such error-prone cues in the presentation and user interface to recorded speech. These acoustically based segmentation methods provide cues that the user can exploit to navigate in, and interactively prune, an acoustical search space.

5.9.1 Recording Issues

SpeechSkimmer was developed on Apple Macintosh computers that include an 8-bit digital-to-analog (D/A) converter for sound output. The hardware supports several sampling rates up to approximately 22 kHz.[53] This maximum sampling rate was used to obtain the best possible sound quality given the hardware limitations. One hour of recorded speech at this sampling rate requires roughly 80 MB of storage.

For recorded speech, a larger dynamic range (i.e., a 12- or 16-bit D/A) will produce better speech quality. A coding scheme such as μ-law can compress approximately 12 bits of dynamic range into 8 bits. Other more complex coding schemes can produce intelligible speech with much larger data compression factors (Noll 1993).

Most of the recordings used in this research were directly recorded on a portable (3 kg) Apple PowerBook computer. Unfortunately, this machine has automatic gain control (AGC) which causes the volume level to automatically increase whenever a recording is started or there is a pause (Keller 1993). AGC is undesirable in these kinds of systems because recordings are created with different gains, complicating speech detection and other forms of acoustic processing.

Three different microphone configurations were used. The Apple omni-directional microphone was used for informal monologues and dialogues. Two pressure zone microphones (Ballou 1987) and a pre-amplifier were used to record classroom discussions. A formal lecture was recorded in a theater by obtaining a feed directly from the main audio mixing board.

5.9.2 Processing Issues

While the intent of this research is to provide real-time interactive skimming capabilities, the retrieval tasks will occur after the creation of a recording. It is therefore practical to perform some off-line analyses of the data. It is not feasible to perform the segmentation on-the-fly in the interactive skimming application, as the entire recording must first be analyzed in order for the adaptive algorithms and segmentation to work.

[53] All sound files contain 8-bit linear samples recorded at 22,254 samples/s.

It is, however, possible to perform the speech detection and pitch-based segmentation at the time of recording, rather than as a post-processing technique.

SpeechSkimmer incorporates several time compression techniques for experimentation and evaluation purposes. Note that these speech processing algorithms run on the main processor of the computer and do not require special signal processing hardware. The current implementation of the sampling technique produces good quality speech and permits a wide range of time compression values. These algorithms run in real time on a Macintosh PowerBook 170 (25 MHz 68030).

An optimized version of the synchronized overlap add technique called SOLAFS (SOLA with fixed synthesis, see Hejna 1990) is also used in SpeechSkimmer. This algorithm allows speech to be slowed down as well as sped up, reduces the acoustical artifacts of the compression process, and provides improved sound quality over the sampling method. The cross correlation of the SOLAFS algorithm performs many multiplications and additions requiring a slightly more powerful machine to run in real time. A Macintosh Quadra 950 (33 MHz 68040) that has several times the processing power of a PowerBook 170 is sufficient.

5.9.3 Speech Detection for Segmentation

An adaptive time-domain speech detector (see chapter 4) was developed for segmenting recordings. The speech detector uses average magnitude and zero crossing measurements combined with heuristic properties of speech. Background noise can then be differentiated from speech under a variety of microphone and noise conditions.

A speech detector based on the work of Lamel et al. (see section 4.4.3) has been developed for pause removal and to provide data for pause-based segmentation. Digitized speech files are analyzed in several passes; the first pass gathers average magnitude and ZCR statistics for 10 ms frames of audio. Note that for most speech recordings these two measurements are relatively independent for large energy and zero crossing values (figure 5-13).

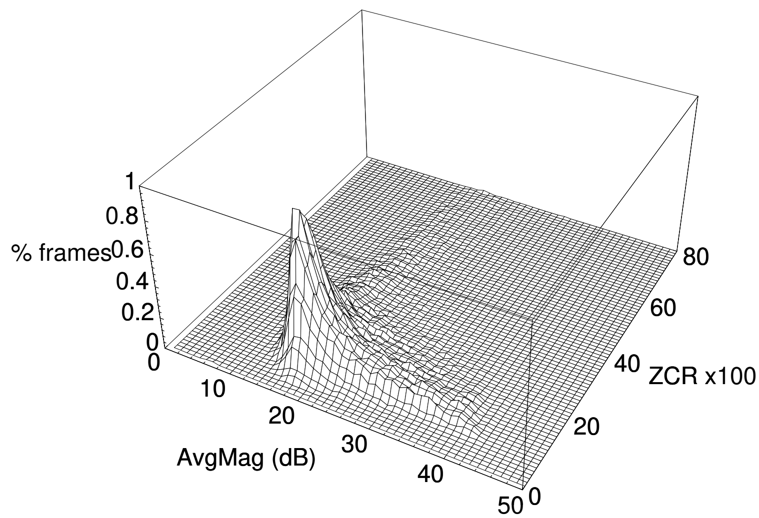


Fig. 5-13. A 3-D plot of average magnitude and zero crossing rate histogram. The data is from a 15 minute recording made in a noisy classroom (10 ms frames).

The background noise level is determined by generating a histogram of the average magnitude measurements and smoothing it with a three-point averaging filter (as in figure 4-5). The resulting histogram typically has a bimodal distribution (figures 5-14 and 5-15); the first peak corresponds to background noise, the second peak to speech. A value 2 dB above the first peak is selected as the initial dividing line between speech and background noise. If it is determined that the overall background noise level is high, a 4 dB offset is used. Figure 5-15 illustrates a recording with a high background level; there are no zero or low energy frames present in the recording, so the speech detector selects the higher offset value.

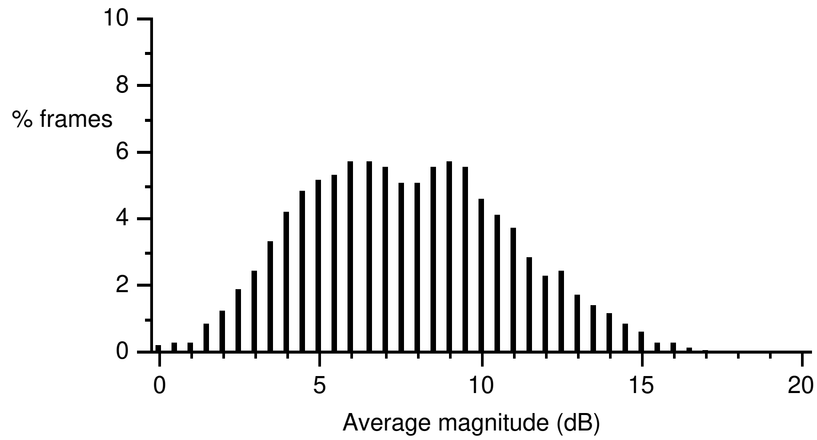


Fig. 5-14. Average magnitude histogram showing a bimodal distribution. The first peak represents the background noise; the second peak represents the speech.

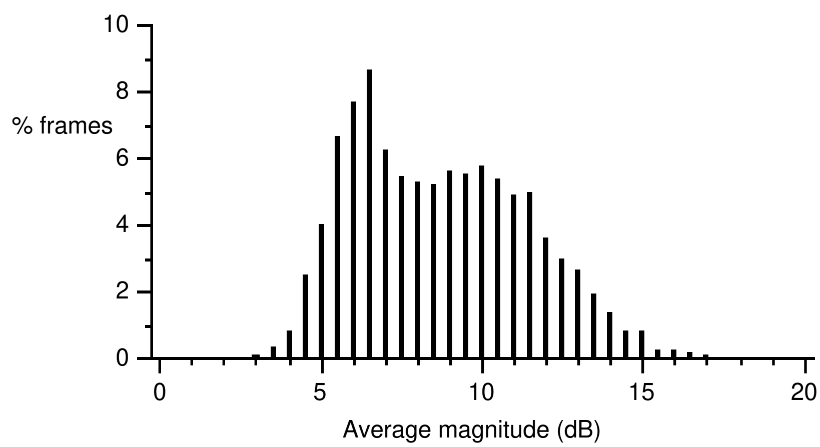


Fig. 5-15. Histogram of noisy recording. Note that there are no zero or low-energy entries.

For determining a zero crossing threshold O'Shaughnessy says:

> A high ZCR cues unvoiced speech while a low ZCR corresponds to voiced speech. A reasonable boundary can be found at 2500 crossings/s, since voiced and unvoiced speech average about 1400 and 4900 crossings/s respectively, with a larger standard deviation for the latter. (O'Shaughnessy 1987, 215)

The noise level, as calculated above, and a ZCR threshold of 2500 crossings/s thus provides an initial classification of each frame as speech or background noise.

Even when the threshold parameters are carefully chosen, some classification errors will be made. However, several additional passes through the sound data are made to refine this estimation based on heuristics of spontaneous speech. This processing fills in short (< 100 ms) gaps between speech segments (see section 4.5 and figure 4-9), removes isolated islands initially classified as speech that are too short to be words (< 100 ms), and extends the boundaries of speech segments (by 20 ms) so that they are not inadvertently clipped (Gruber 1982; Gruber 1983). For example, two or three frames (20–30 ms) initially classified as background noise amid many high-energy frames identified as speech should be treated as part of that speech, rather than as a short interposing silence. Similarly, several medium-energy frames in a large region of silence are too short to be considered speech and are filtered out to become part of the silence.

This speech detection technique has been found to work well under a variety of noise conditions. Audio files recorded in an office environment with computer fan noise and in a lecture hall with over 40 students have been successfully segmented into speech and background noise. This pre-processing of a sound file executes in faster than real time on a personal computer (e.g., it takes roughly 30 seconds to process a 100 second sound file on a PowerBook 170).

Several changes were made to the speech detector as the skimming system evolved. Preliminary studies were performed on a Sun SparcStation using μ-law speech that encodes approximately 12 bits of dynamic range. It was easy to differentiate the peaks in the bimodal energy distribution in histograms made with bins 1 dB wide. Note that averaging the magnitude or energy values over 10 ms frames reduces the effective dynamic range for the frames. The Macintosh only encodes 8 bits of dynamic range, and with 1 dB wide bins it was sometimes difficult to distinguish the two modes of the distribution. Using a smaller bin size for the histogram (i.e., 0.5 dB) made it easier to differentiate the peaks. For example, note that the modes in figure 5-14 may be hard to find if the bins were 1 dB wide.

Some of the speech recordings used in this research were created in a theater through the in-house sound system. The quality of this recording is very good, but it contains some high frequency noise from the amplification equipment. This noise resulted in a high zero crossing rate and hence incorrectly classified background noise as speech. This recording was low-pass filtered to remove frequency components above about 5 kHz to eliminate the false triggering of the speech detector. Once the speech detector has been run, the original unfiltered recording is used for playback to produce the best possible sound quality. Note that such low pass filtering can be blindly applied to all sound files without affecting the results.

The speech detector outputs an ASCII file containing the starting time, stopping time, and duration of each segment,[54] and a flag indicating if the segment contains speech or background noise (figure 5-16).

	4579	4760	181	0
	4760	5000	240	1
	5000	5371	371	0
	5371	5571	200	1
	5571	6122	551	0
	6122	6774	652	1
➔	6774	7535	761	0
	7535	7716	181	1
	7716	7806	90	0
	7806	9509	1703	1
	9509	9730	221	0
	9730	9900	170	1
	9900	10161	261	0
	10161	10391	230	1
	10391	10541	150	0
	10541	11423	882	1
	11423	11534	111	0
	11534	12245	711	1
	12245	12395	150	0

Fig. 5-16. Sample speech detector output. Columns are: start time, stop time, duration of segment, and speech present (1=speech, 0=background noise). Note the long pause that occurs at 6674 ms. All times are in milliseconds.

5.9.4 Pause-based Segmentation

Even though this software is designed to run on a Macintosh, a UNIX tools approach is taken in the design of the system to simplify the software components (Kernighan 1976). Segmentation into salient segments is run as a second process on the speech detection data shown in figure 5-16. This modularity allows for experimentation with different pause-based segmentation algorithms on the raw speech detection data. See section 5.11 for how these data are used in the interactive system.

[54]The duration field is not necessary, but has been found useful for visual debugging.

	4760	5370	1	1
	5371	5570	2	0
	5371	6121	1	0
	5622	6121	2	0
	6122	6773	2	0
	6122	7534	1	0
	7035	7534	2	0
	7535	7715	2	0
➔	7535	7715	3	1
	7535	7805	1	0
	7806	9508	2	1
	7806	9508	3	0
	7806	9729	1	1
	9730	9899	2	0
	9730	9899	3	0
	9730	10160	1	0
	10161	10390	2	0
	10161	10390	3	0
	10161	10540	1	0
	10541	11422	2	0
	10541	11422	3	0
	10541	11533	1	0
	11534	12244	2	0
	11534	12244	3	0
	11534	12394	1	0

Fig. 5-17. Sample segmentation output. Columns are: start time, stop time, skimming level, and the jump-to flag (1=OK to jump to). Note the correspondence between these data and figure 5-16. A valid starting segment for level 3 skimming occurs here at 7353 ms, this occurs just after the long (761 ms) silence in figure 5-16 beginning at 6774 ms.

5.9.5 Pitch-based Emphasis Detection for Segmentation

Pitch[55] provides information in speech that is not only important for comprehension and understanding but can also be exploited for machine-mediated systems. There are many techniques to extract pitch (Hess 1983; O'Shaughnessy 1987; Keller 1992), but there have been few attempts to extract high-level information from the speech signal based on pitch.

Work in detecting emphasis (Chen 1992), locating intonational features (Hirschberg 1987, Wightman 1992), and finding syntactically significant hesitations based on pause length and pitch (O'Shaughnessy 1992) has just begun to be applied to speech segmentation and summarization. SpeechSkimmer builds upon these ideas and is structured to integrate this type of information into an interactive interface.

[55]"Pitch" in this context means the fundamental frequency of voiced speech, and is often denoted as F0. The terms "pitch," "fundamental frequency," and "F0" are used interchangeably in this document.

Chen and Withgott (Chen 1992) trained a Hidden Markov Model (HMM, see Rabiner 1989) to detect emphasis based on hand-marked training data of the pitch and energy content of conversations. Emphasized portions in close temporal proximity were found to successfully create summaries of the recordings. This prosodic approach is promising for extracting high-level information from speech signals. While HMMs are well understood in the speech recognition community, they are computationally complex statistical models that require significant amounts of training data.

While performing some exploratory data analysis on ways to improve on this HMM-based approach, it became clear that the fundamental frequency of speech itself contains emphasis information. Rather than collecting a large amount of training for an HMM, it appeared possible to create a much simpler emphasis or structure detector by directly looking for patterns in the pitch.

For example, it is well known in the speech and linguistics communities that there are changes in pitch under different speaking conditions (Hirschberg 1992; Hirschberg 1986; Silverman 1987). The introduction of a new topic often corresponds with an increased pitch range. There is a "final lowering," or general declination of pitch during the production of a sentence. Sub-topics and parenthetical comments are often associated with a compression of pitch range. Such pitch features are reasonably robust within and across native speakers of American English.[56]

These are general rules of thumb, however, automatically finding these features in a speech signal is difficult as the actual pitch data tends to be noisy. Several techniques were investigated to directly find such features in a speech signal (e.g., fitting the pitch data to a curve or differencing the endpoints of contiguous segments); however the pitch data was noisy and the features of interest were difficult to find in a general manner.

Several experiments were performed by visually correlating areas of activity in an F0 plot with a hand-marked log of a recording. Areas of high pitch variability were strongly correlated with new topic introductions and emphasized portions of the log. Visually it is easy to locate areas of significant pitch activity (figure 5-18); however, if we could write a program to extract features that are easy to see with our visual system, we would have been able to solve the much larger and more difficult problem of image understanding.

Figure 5-18 shows the F0 for 40 seconds of a recorded monologue. There are several clearly identifiable areas of increased pitch activity. Figure 5-

[56]Pitch is used differently in other languages, particularly "tone languages" where pitch is used phonemically (i.e., to distinguish words).

19 is a close-up of several seconds of the same data. Note that the pitch track is not continuous; pitch can only be calculated for vowels and voiced consonants (e.g., "v," "z"); consonants such as "s," "f," "p" are not voiced. Also note that pitch extraction is difficult (Hess 1983; Keller 1992), the resulting data is noisy, and contains anomalous points.

A variety of statistics were generated and manually correlated with the hand-marked log. The statistics were gathered over one second windows of the pitch data (100 frames of 10 ms). One second was chosen to aggregate a reasonable number of pitch values, and to correspond with the length of several words. The metrics evaluated include the mean, standard deviation, minimum, maximum, range, number of frames above a threshold, and number of local peaks, across the one second window.

The range, maximum, standard deviation, and number of frames above a threshold were most highly correlated with the hand-marked data. The standard deviation and number of frames above a threshold appear the most promising for emphasis detection and summarization. Note that all these metrics essentially measure the same thing: significant activity and variability in F0 (figure 5-21).

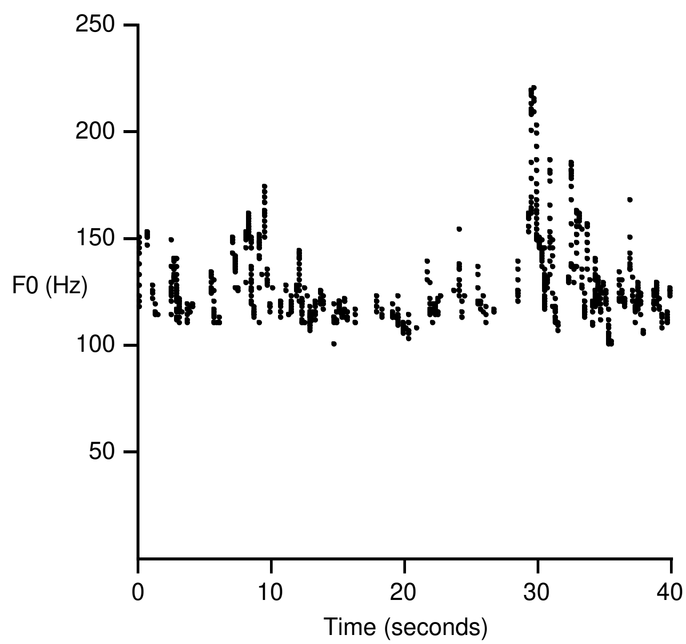


Fig. 5-18. F0 plot of a monologue from a male talker. Note that the area near 30 seconds appears (and sounds) emphasized. The F0 value is calculated every 10 ms.

Since the range and baseline pitch vary considerably between talkers, it is necessary to analyze the data to find an appropriate threshold for a particular talker. A histogram of the F0 data is used, and a threshold is chosen to select the top 1% of the pitch frames (figure 5-20).[57]

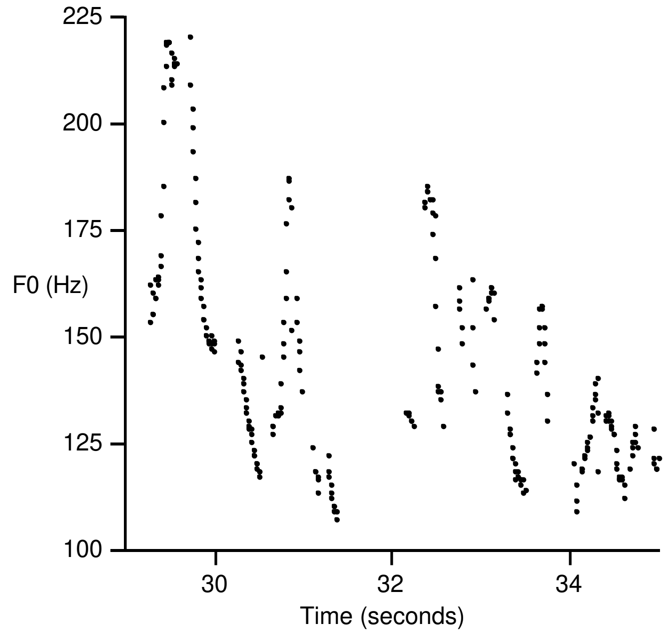


Fig. 5-19. Close-up of F0 plot in figure 5-18.

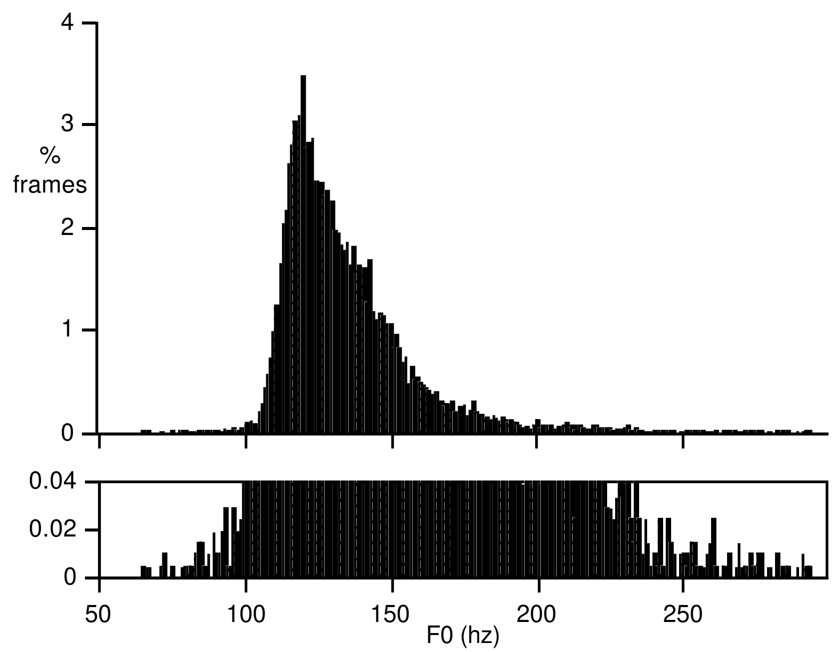


Fig. 5-20. Pitch histogram for 40 seconds of a monologue from a male talker. The bottom portion of the figure shows a greatly expanded vertical scale illustrating the occurrence of pitch frames above 200 Hz.

[57]This threshold was chosen as a practical starting point. The threshold can be changed to find a larger or smaller number of emphasized segments.

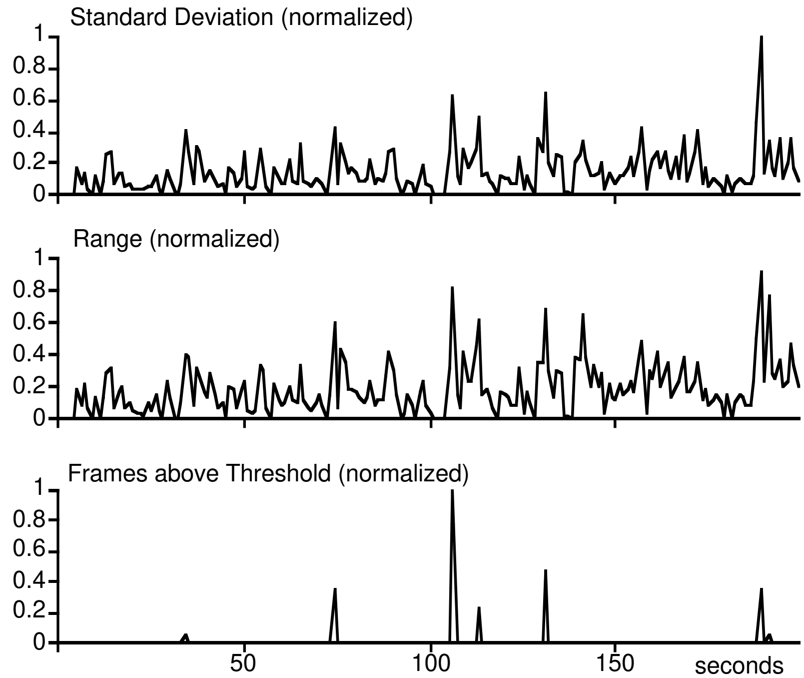


Fig. 5-21. Comparison of three F0 metrics.

The number of frames in each one second window that are above the threshold are counted as a measure of "pitch activity." The scores of nearby windows (within an eight second range) are then combined. For example, a speech activity of four in window 101 (i.e., four frames above the threshold) would be added to a speech activity of three in frame 106 to indicate there is a pitch activity of seven for the region of 101–108 seconds. This method is used instead of analyzing eight second windows so that the start of the pitch activity can be found at a finer (one second) granularity.

5.10 Usability Testing

The goal of this test was to find usability problems as well as areas of success in the SpeechSkimmer system.[58] The style of usability test performed is primarily an observational "thinking out loud" study (Ericsson 1984) that is intended to quickly find major problems in the user interface to an interactive system (Nielsen 1993a).

[58]This test was conducted under the guidelines of, and approved by, the M.I.T. Committee on the Use of Humans as Experimental Subjects (application number 2132).

5.10.1 Method

5.10.1.1 Subjects

Twelve volunteer subjects between the ages of 21 and 40 were selected from the Media Laboratory environment.[59] None of the subjects were familiar with the SpeechSkimmer system, but all had experience using computers. Six of the subjects were graduate students and six were administrative staff; eight were female and four were male. Test subjects were not paid, but were offered snacks and beverages to compensate for their time.

5.10.1.2 Procedure

The tests were performed in an acoustically isolated room with a subject, an interviewer, and an observer.[60] The sessions were video taped and later analyzed by both the interviewer and observer. A testing session took approximately 60 minutes and consisted of five parts:

1. A background interview to collect demographic information and to determine what experience subjects had with recorded speech and audio. The questions were tailored to the subject's experiences. For example, someone who regularly recorded lectures would be asked in detail about their use of the recordings, how they located specific pieces of information in the recordings, etc.

2. A first look at the touchpad. Subjects were given the touchpad (figure 5-10) and asked to describe their first intuitions about the device. This was done without the interviewer revealing anything about how the system worked or what it is intended to do, other than "it is used for skimming speech recordings." Everything subjects did in the test was exploratory, they were not given any instructions or guidance.[61] The subjects were asked what they thought the different regions of the device did, how they expected the system to behave, what they thought backward did, etc.

3. Listening to a trial speech recording with the SpeechSkimmer system. The subjects were encouraged to explore and "play" with the device to confirm, or discover, how the system operated. While investigating the device, the interviewer encouraged the subjects to "think aloud," to

[59]One subject was a student visiting the lab, another was a temporary office worker.
[60]L. Stifelman conducted the test, the system designer (Arons) observed.
[61]However, if a subject said something like "I wish it did X," and the system did perform that function, the feature was revealed to them by the interviewer through directed questions (e.g., Do you think this device can do that? If so, how do you think you could get it to do it? What do you think that button does?).

describe what they were doing, and to say if the device was behaving as they had expected.

4. A skimming comparison and exercise. This portion of the test compared two different skimming techniques. A recording of a 40 minute lecture was divided into two 20 minute parts.[62] Each subject listened to both halves of the recording; one part was segmented based on pitch (section 5.9.5) and one that was segmented isochronously (at equal time intervals). The test was counterbalanced for effects of presentation order and portion of the recording (figure 5-22).

# of subjects	first presentation		second presentation	
3	pitch-based	part 1	isochronous	part 2
3	isochronous	part 1	pitch-based	part 2
3	isochronous	part 2	pitch-based	part 1
3	pitch-based	part 2	isochronous	part 1

Fig. 5-22. Counterbalancing of experimental conditions.

When skimming, both of the techniques provided a 12:1 compression for this recording (i.e., on average five seconds out of each minute were presented). Note that these figures are for normal speed (1.0x), by using time compression the subjects could achieve over a 25:1 time savings.

The subjects first skimmed the entire recording at whatever speed they felt most comfortable. The subjects were asked to judge (on a 7-point scale) how well they thought the skimming technique did at providing an overview of the recording and selecting indexes into major points in the recording. The subjects were then given a printed list of three questions that could be answered by listening to the recording. The subjects were asked locate the answer to any of the questions in the recording, and to describe their auditory search strategy. This process was repeated for the second presentation condition.

5. The test concluded with follow-up questions regarding the subject's overall experience with the interaction device and the SpeechSkimmer system, including what features they disliked and liked most.

5.10.2 Results and Discussion

This section summarizes the features of the SpeechSkimmer system that were frequently used or liked the most by the subjects of the usability test, as well as areas for improvement in the user interface design.

[62]The recording is of Nicholas Negroponte's "Perspectives Speaker Series" talk titled *Conflusion: Media in the Next Millennium* presented on October 19, 1993.

5.10.2.1 Background Interviews

All the subjects had some experience in searching for recorded audio information on compact discs, audio cassettes, or video tape. Subjects' experience included transcribing lectures and interviews, taking personal notes on a microcassette recorder, searching for favorite songs on tape or CD, editing video documentaries, and receiving up to 25 voice mail messages per day. Almost all the subjects referred to the process of searching as time consuming, one subject added that it takes "more time than you want to spend."

5.10.2.2 First Intuitions

Most of the users found the interface intuitive and easy to use, and were able to use the device without any training. This ability to quickly understand how the device works is partially based on the fact that the touchpad controls are labeled in a similar manner as consumer devices (such as compact disc players and video cassette recorders). While this familiarity allowed the subjects to initially feel comfortable with the device, and enabled rapid acclimatization to the interface, it also caused some confusion since a few of the functions behaved differently than on the consumer devices.

Level 2 on the skimming template is labeled "no pause" but most of the subjects did not have any initial intuitions about what it meant. The label baffled most of the subjects since current consumer devices do not have pause removal or similar functionality. Some of the subjects thought that once they started playing in "no pause" they would not be able to stop or pause the playback. Similarly, the function of the "jump and play normal button" was not obvious. Also, the backward play levels were sometimes intuitively equated with traditional (unintelligible) rewind.

5.10.2.3 Warm-up Task

The recording used in the trial task consisted of a loose free-form discussion, and most of the subjects had trouble following the conversation. Most said that they would have been able to learn the device in less time if the trial recording was more coherent, or if they were already familiar with the recording. However, subjects still felt the device was easy to learn quickly.

Subjects were not sure how far the jumps took them. Several subjects thought that the system jumped to the next utterance of the male talker when exploring the interface in the trial task (the first few segments selected for jumping in this recording do occur at a change of talker).

5.10.2.4 Skimming

Most subjects thought, or found, that the pitch-based skimming was effective at extracting interesting points to listen to, and for finding information. One user who does video editing described it as “grabbing sound bite material.” When comparing pitch-based skimming to isochronous skimming a subject said “it is like using a rifle versus a shotgun” (i.e., high accuracy instead of dispersed coverage). Other subjects said that the pitch-based segments “felt like the beginning of phrase … [were] more summary oriented” and there was “a lot more content or keyword searching going on” than in the isochronous segmentation.

A few of the subjects requested that longer segments be played (perhaps until the next pause), or that the length of the segments could be controllable. One subject said “I felt like I was missing a lot of his main ideas, since it would start to say one, and then jump.”

The subjects were asked to rank the skimming performance under the different segmentation conditions. A score of 7 indicates the best possible summary of high-level ideas, a score of 1 indicates very poorly selected segments. The mean score for the pitch-based segmentation was $M = 4.5$ ($SD = 1.7$, $N = 12$); the mean score for the isochronous segmentation was $M = 2.7$ ($SD = 1.4$, $N = 12$). The pitch-based skimming was rated better than isochronous skimming with a statistical significance of $p < .01$ (using a t test for paired samples). No statistically significant difference was found on how subjects rated the first versus the second part of the talk, or on how subjects rated the first versus second sound presented.

Most of the subjects, including the few that did not think the pitch-based skimming gave a good summary, used the skimming level to navigate through the recording. When asked to find the answer to a specific question, most started off by saying something like “I’ll go the beginning and skim till I get to the right topic area in the recording,” or in some cases “I think its near the end, so I’ll jump to the end and skim backward.”

5.10.2.5 No Pause

While there was some initial confusion regarding the “no pause” level, if a subject discovered its function, it often became a preferred way to quickly listen and search for information. One subject that does video editing said “that’s nice … I like the no pause function.… it kills dead time between people talking … this would be really nice for interviews [since you normally have to] remember when he said [the point of interest], then you can't find where it was, and must do a binary search of the audio track … For interviews it is all audio—you want to get the sound bite."

5.10.2.6 Jumping

The function of the "jump and play normal" button was not always obvious, but subjects who did not initially understand what the button did found ways to navigate and perform the same function using the basic controls. This button is a short-cut: a combination of jumping backward and then playing level 1 speech at regular speed.

One subject had a moment of inspiration while skimming along at a high speed, and tried the button after passing the point of interest. After using this button the subject said in a confirming tone "I liked that, OK." The subject proceeded to use the button several times after that and said "now that I figured out how to do that jump normal thing … that's very cool. I like that." It is important to note that after discovering the "jump and play normal" button this subject felt more comfortable skimming at faster speeds. Another subject said "that's the most important button if I want to find information."

While most of the subjects used, and liked, the jump buttons, the size or granularity of jumps was not obvious. Subjects assumed that jumping always brought them to the next sentence or topic.[63] While using the jump button and "backward no pause" one subject noted "oh, I see the difference … I can re-listen using the jump key."

5.10.2.7 Backward

Most of the subjects figured out the backward controls during the trial test, but tended to avoid using them. This is partially attributable to the subject's initial mental models that associate backward with "unintelligible" rewind. Some of the subjects, however, did find the backward levels useful in locating particular words or phrases that had just been heard.

While listening to the recording played backward, one subject noted "it's taking units of conversation—and goes backwards." Another subject said that "it's interesting that it is so seamless" for playing intelligible segments and that "compared to a tape where you're constantly shuffling back and forth, going backward and finding something was much easier

[63] In the current system design the amount of the jumps depends on the current level (normal, no pause, or skimming).

since [while] playing backwards you can still hear the words." One subject suggested providing feedback to indicate when sounds were being played backward, to make it easily distinguishable from forwards.

5.10.2.8 Time Compression

Some of the users thought there were only three discrete speeds and did not initially realize that there was a continuum of playback speeds. A few of the subjects also did not initially realize that the ability to change speeds extended across all the skimming levels. These problems can be attributed to the three speeds marked on the template (slow, regular, and fast, see figure 5-11). One subject noted that the tactile strips on the surface break the continuity of the horizontal "speed" lines, and made it less clear that the speeds work at all skimming levels.[64]

Several of the subjects thought there was a major improvement when listening over the headphones, one subject was "really amazed" at how much better the dichotic time-compressed speech was for comprehension than the speech presented over the loudspeaker. Another subject said "it's really interesting—you can hear it a lot better."

5.10.2.9 Buttons

The buttons were generally intuitive, but there were some problems of interpretation and accidental use. The "begin" and "end" regions were initially added next to the level 3 and –3 skimming regions on the template to provide a continuum of playback granularity (i.e., normal, no pause, skim, jump to end). Several subjects thought that the begin button should seek to the beginning of the recording and then start playing.[65] One subject additionally thought the speed of playback could be changed by touching at the top or bottom of the begin button.

One subject wanted to skim backward to re-hear the last segment played, but accidentally hit the adjacent begin button instead. This frustrated the subject, since the system jumped to the beginning of the recording and hence lost the location of interest.

It should also be noted that along with these conceptual and mechanical problems, the words "begin" and "start" are overloaded and could mean "begin playing" as well as "seek to the beginning of the recording."

---

[64]Two of the subjects suggested using colors to denote the continuum of playback speeds and that the speed labels extend across all the skimming levels.
[65]The system seeks to the beginning of the recording and then pauses.

By far the biggest problem encountered during the usability test was "bounce" on the jump and pause buttons.[66] This was particularly aggravating when it occurred with the pause button, as the subject would want to stop the playback, but the system would temporarily pause, then moments later un-pause. The bounce problem was partially exacerbated by the subject's use of their thumbs to touch the buttons. While the touchpad and template were designed to be operated with a single finger for maximum dexterity and accuracy (as in figure 5-10), most of the subjects held the touchpad by the right and left sides and touched the surface with their thumbs during the test.[67]

5.10.2.10 Non-Speech Feedback

The non-speech audio was successful at unobtrusively providing feedback. One subject, commenting on the effectiveness and subtlety of the sounds said "after using it for a while, it would be annoying to get a lot of feedback." Another subject said that the non-speech audio "helps because there is no visual feedback." None of the subjects noted that the frequency of the feedback tone changes with skimming level; most did not even notice the existence of the tones. However, when subsequently asked about the device many noted that the tones were useful feedback to what was going on. The cartoon "boings" at the beginning and ending were good indicators of the end points (one subject said "it sounds like you hit the edge"), and the other sounds were useful in conveying that something was going on. The boing sounds were noticed most often, probably because the speech playback stops when the sound effect is played.

5.10.2.11 Search Strategies

Several different navigation and search strategies were used to find answers to specific questions within the recordings. Most of the subjects skimmed the recording to find the general topic area of interest, then changed to level 1 playing or level 2 with pauses removed, usually with time compression. One subject started searching by playing normally (no time compression) from the beginning of the recording to "get a flavor" for the talk before attempting to skim or play it at a faster rate. One subject used a combination of skimming and jumping to quickly navigate through a recording and efficiently find the answers to specific questions.

---

[66]Button "bounce" is traditionally associated with mechanical switches that would make several temporary contact closures before settling to a quiescent state. The difficulties here are associated with the way in which the touchpad is configured.

[67]This was partially attributable to the arrangement of the subject and the experimenters during the test. There was no table on which to place the touchpad, and subjects had to hold the device.

5.10.2.12 Follow-up Questions

Most of the subjects thought that the system was easy to use since they made effective use of the skimming system without any training or instructions. Subjects rated the ease of use of the system on a 7-point scale where 1 is difficult to use, 4 is neutral, and 7 is very easy to use. The mean score for ease of use was $M = 5.4$ ($SD = 0.97$, $N = 10$).[68]

Most subjects liked the ability to quickly skim between major points in a presentation, and to jump on demand within a recording. Subjects liked the time compression range, particularly the interactive control of the playback speed. A few of the subjects were enamored with other specific features of the system including the “fast-forward no pause” level, the “jump and play normal” button, and the dichotic presentation.

One subject commented “I really like the way it is laid out. It’s easier to use than a mouse.” Another subject (who did not realize the speeds were continuous) experimented with turning the touchpad 90 degrees so that moving a finger horizontally rather than vertically changed the playback speed.

Most of the subjects said they could envision using the device while doing other things, such as walking around, but few thought they would want to use it while driving an automobile. Most of the subjects said they would like to use such a device, and many of them were enthusiastic about the SpeechSkimmer system.

5.10.2.13 Desired Functionality

In the follow-up portion of the test, the subjects were asked what other features might be helpful for the speech skimming system. For the most part these items were obtained through probing the test subject, and were not spontaneously mentioned by the subjects.

Some subjects were interested in marking points in the recording that were of interest to them, for the purpose of going back later and to access those points. A few of the subjects called these “bookmarks.”

Some subjects wanted to be able to jump to a particular place in a recording, or have a graphical indicator of their current location. There is a desire, for example, to access a thought discussed “about three-quarters the way through the lecture” through using a “time line” for jumping within a recording.

---

[68] Two of the subjects did not answer the question.

5.10.3 Thoughts for Redesign

After establishing the basic system functionality, the touchpad template evolved quickly—figure 5-23 shows three prototype templates as well as the current design. It is important to note again that this usability test was performed without any instruction or coaching of the subjects. It may be easy to fix some, or most, of these problems through a small amount of instruction, or by modifying the touchpad template.

A revised template can alleviate some of the usability problems encountered and incorporate the new features requested. The "sketch" in figure 5-24 shows a prototype of a new design. The labels and icons are modified to be more consistent and familiar. Notably, "play" has replaced "normal," and "pauses removed" has replaced the confusing "no pause."

The speed labels are moved, renamed, and accompanied by tick marks to indicate a continuum of playback rates. The shaded background is as an additional cue to the speed continuum that extends across all levels. Colors, however, may be more effective than of shading. For example, the slow-to-normal range could fade from blue to white, while the normal-to-fastest range could go from white to red, suggesting a cool-to-hot transition.

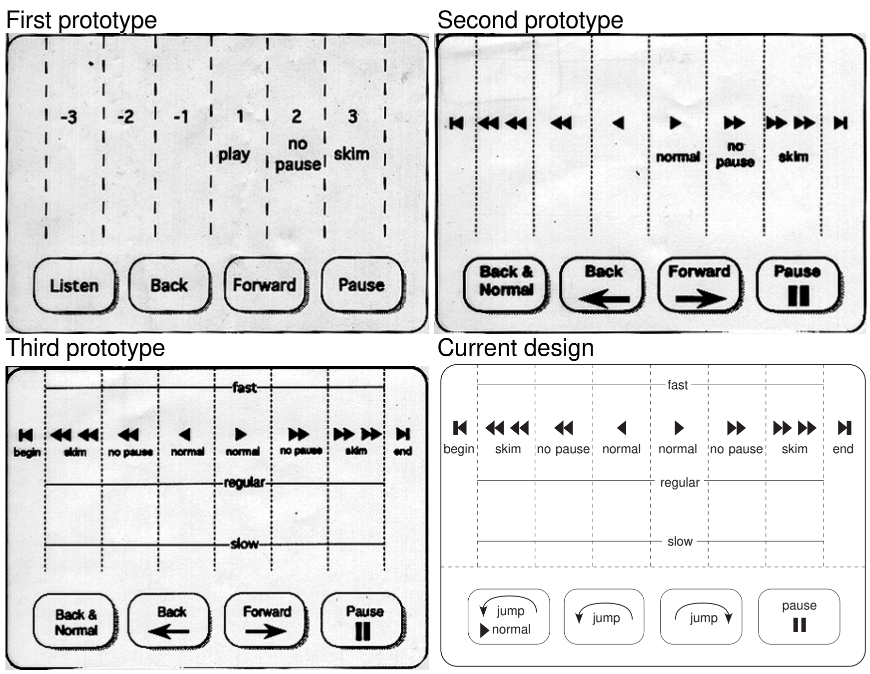


Fig. 5-23. Evolution of SpeechSkimmer templates.

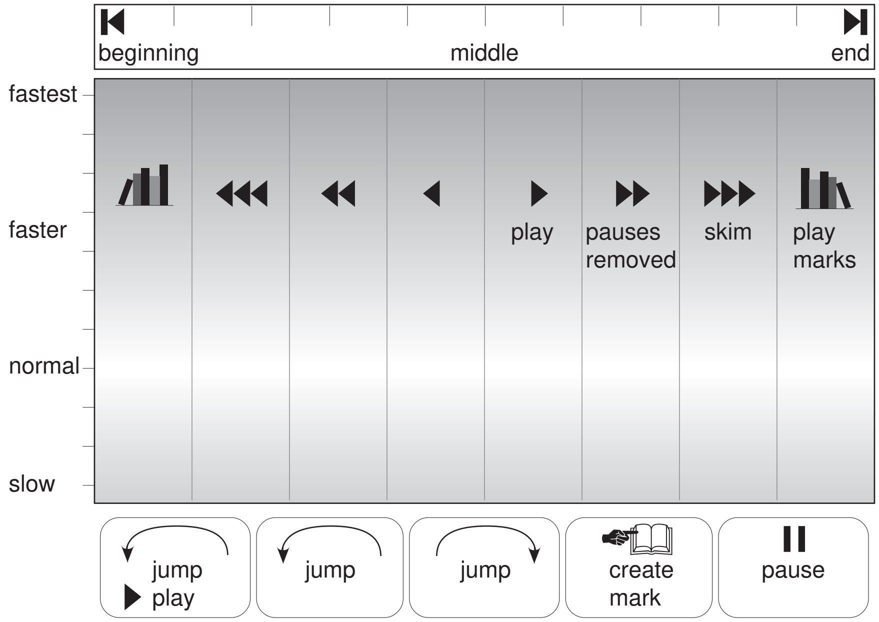


Fig. 5-24. Sketch of a revised skimming template.

Bookmarks, as requested by the subjects, can be implemented in a variety of ways, but are perhaps best thought of as yet another level of skimming. In this case, however, the user interactively selects the speech segments on-the-fly. In this prototype a "create mark" button is added along with new regions for playing forward and backward between the user defined marks.

A time line is added to directly access time points within a recording. It is located at the top of the template where subjects pointed when talking about the feature. The time line also naturally incorporates the begin and end controls, removing them from the main portion of the template and out of the way from accidental activation.

There is room for improvement in the layout and graphic design of this template, it is somewhat cluttered, and the "jump and play normal" button remains problematic. However, the intuitiveness of this prototype, or alternative designs, could be quickly tested by asking a few subjects for their initial impressions.

One of the subjects commented that a physical control (such as real buttons and sliders) would be easier to use than the touchpad. A slightly different approach to changing the physical interface to the skimming system is to use a jog and shuttle control, as is often found in video editing systems (figure 5-25). Alternatively, a foot pedal could be used in situations where the hands are busy, such as when transcribing or taking notes.

Fig. 5-25. A jog and shuttle input device.

5.10.4 Comments on Usability Testing

Informal heuristic evaluation of the interface (Nielsen 1990; Nielsen 1991; Jeffries 1991) was performed throughout the system design. In addition, the test described in section 5.10 was very helpful in finding usability problems. The test was performed relatively late in the SpeechSkimmer design cycle, and, in retrospect, a preliminary test should have been performed much earlier in the design process. Most of the problems in the template layout could have been uncovered earlier, with only a few subjects. This could have led to a more intuitive interface, while focusing on features that are most desired by users.

Note that while twelve subjects were tested here, only a few are needed to get helpful results. Nielsen has shown that maximum cost-benefit ratio for a usability project occurs with around three to four test subjects, and that even running a single test subject is beneficial (Nielsen 1993b).

5.11 Software Architecture

The software implementation consists of three primary modules: the main event loop, the segment player, and the sound library (figure 5-26). The skimming user interface is separated from the underlying mechanism that presents the skimmed and time-compressed speech. This modularization allows for the rapid prototyping of new interfaces using a variety of interaction devices. SpeechSkimmer is implemented using objects in THINK C 5.0, a subset of C++.[69]

[69]Think C 5.0 provides the object-oriented features of C++, but does not include other extensions to C such as operator overloading, in-line macros, etc.

The main event loop gathers raw data from the user and maps it onto the appropriate time compression and skimming ranges for the particular input device. This module sends simple requests to the segment player to set the time compression and skimming level, start and stop playback, and jump to the next segment.

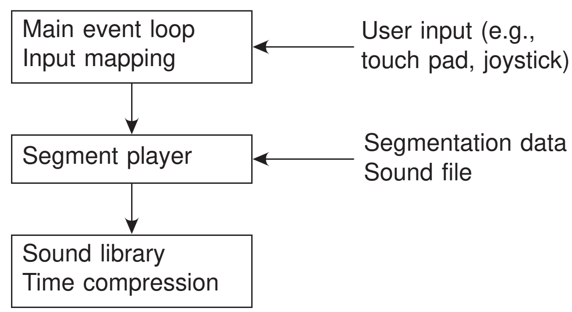


Fig. 5-26. Software architecture of the skimming system.

The segment player is the core software module; it combines user input with the segmentation data to select the appropriate portion of the sound to play. When the end of a segment is reached, the next segment is selected and played. Audio data is read from the sound file and passed to the sound library. The size of these audio data buffers is kept to a minimum to reduce the latency between user input and the corresponding sound output.

The sound library provides a high-level interface to the audio playback hardware (based on the functional interface described in Arons 1992c). The time compression algorithms (Fairbanks sampling, dichotic sampling, SOLAFS) are built into the sound library.

5.12 Use of SpeechSkimmer with BBC Radio Recordings

A related project in the Speech Research Group at the Media Laboratory is concerned with the structuring and presentation of news stories collected from broadcast radio. News broadcasts, at least those produced by National Public Radio and the BBC, are much more structured than the recordings of spontaneous speech discussed in this document. For example, BBC News Hour broadcasts contain summaries of the main points of the news at the beginning and end of the broadcast, there is often a change of announcer between story segments, and it is possible to find certain story boundaries by looking for musical segues (Hawley 1993). A careful analysis of such broadcasts has enabled the design of a BBC-specific segmenter that finds story boundaries based on the length and location of pauses in a typical broadcast.

Data collected from this application has been integrated into the SpeechSkimmer framework. A recording of the BBC News Hour is processed by both the BBC-specific segmenter and the pause-based segmenter described in section 5.9.4. The BBC-specific segmentation data is used in place of the level 3 segments. This scheme allows one to interactively listen and skim between news stories using the SpeechSkimmer system.

In the process of integrating the BBC-specific data into the speech skimming system, a fellow graduate student manually segmented a ten minute BBC recording. This effort provides anecdotal evidence to support the effectiveness of the pause-based segmentation system developed in this dissertation. The student spent roughly 45 minutes with a graphical sound editor attempting to find the story boundaries. The recording was then processed by the software described in section 5.9.4. According to the student there was "a very close correspondence between the manual and automatic segmentation" and the segmentation software "did a great job of finding the story boundaries."

5.13 Summary

The SpeechSkimmer system is designed around the premise that navigating in time is critical in speech systems. This chapter has presented a system that integrates time compression, selective pause removal, and perceptually salient segmentation into an interactive interface for presenting and skimming recorded speech.

The system demonstrates that simple heuristics can be powerful for segmenting and listening to recorded speech. For example, the speech that occurs after long pauses can be used as an indicator of structural information conveyed by the talker. Pitch information can provide a more powerful cue to the structure and semantic content of our speech. This chapter describes methods to extract these types of information through simple and efficient measures. Automatic recognition of these structural features may fail by missing things that are important and finding things that are not. However, the interface to the system allows the user to navigate around and through these types of errors. SpeechSkimmer allows intelligent filtering and presentation of recorded audio—the intelligence is provided by the interactive control of the user.

6 Conclusion

This final chapter presents areas for continued research and some concluding thoughts on interactively skimming speech recordings.

6.1 Evaluation of the Segmentation

Early in the design process of the speech skimming system, a variety of speech data were segmented based on pauses. These data included a five minute conversation recorded in an office environment, several 20 minute portions of a classroom discussion recorded in a relatively noisy lecture hall, and many test monologues recorded in an acoustically isolated room. Most of these recordings were made with a low-cost microphone designed for use with the Macintosh. After an unsatisfactory initial test using a single microphone, the classroom lectures were ultimately recorded with two pressure zone microphones (see section 5.9.1 for further details on the recording process).

SpeechSkimmer was demonstrated and tested informally using these recordings. The pause-based segmentation was effective at providing an index into important points in the recordings (also see section 5.12). In both the recorded conversation and the classroom discussions, for example, many of the automatically selected segments corresponded to new topics and talker changes. Some uninteresting, or seemingly random segments, were mixed in with these segments, but these were easy to skip over by using the interactive interface.

The initial investigation of pitch-based segmentation was made on higher quality recordings that were created during an "off-site" workshop. Approximately fifteen talkers introduced themselves, and presented a 10–15 minute summary of their background and interests. These monologues were recorded with a lavaliere microphone on a digital audio tape recorder (2 channels of 16-bit data sampled at 48 kHz).[70]

The first half of one of the monologues (of a male talker) was analyzed while developing the pitch-based segmentation. This entire recording

[70] A "shotgun" microphone was used to record comments and questions from the audience on the other audio channel.

along with two of the other monologues (one male and one female talker) were then segmented using the technique described in section 5.9.5. The portions selected from the second half of the recording were highly correlated with topic introductions, emphasized phrases, and paragraph boundaries in an annotated transcript of the recording.[71]

The four highest scoring segments (i.e., the most pitch activity above the threshold) of each of these recordings were then informally evaluated. People that hear these selected segments generally agree that they are emphasized points or introductions of new topics. The four highest ranking segments[72] for one of the talkers are:

> OK, so the network that we're building is [pause]. Well this [diagram] is more the VuStation, but the network …
>
> OK, the second thing I wanted to mention was, the multimedia toolkit. And currently this pretty much something runs on a …
>
> Currently I'm interested in [pause] computer vision, because I think …
>
> And then, the third program which is something my group is very interested in and we haven't worked on a lot, is the idea of a news parser …

Along with the stated topic introductions, note the inclusion of the linguistic cue phrases "OK" and "so" that are often associated with new topics (Hirschberg 1987).

The pitch-based segmentation technique was applied to a 40 minute lecture for the usability test (section 5.10).[73] The majority of the automatically selected segments were interesting and, as described in section 5.10.2.12, subjects rated the performance of the pitch-based segmentation higher than the isochronous segmentation. Seven of the twelve subjects (58%) gave the pitch-based skimming a rating of 5 or greater for how well it summarized and provided important points in the recording.

From these experiments and evaluations it is believed that both the pause-based and pitch-based techniques are effective at finding relevant segments in speech recordings. The pitch-based technique is currently favored for more effectively selecting salient segments. While some errors are made (selecting unimportant portions, and missing important

[71]The transcript and annotations were independently created by an experienced linguist.
[72]These segments represent eight seconds of the original recording.
[73]Note that many of the selected segments of this recording also contain linguistic cue phrases.

ones), they are easily navigated around and through using the interactive interface, letting the user find, and listen to, things they are interested in.

6.2 Future Research

While graphical user interfaces are concerned with issues of “look and feel,” speech interfaces often have a much different flavor. The “sound and feel” of SpeechSkimmer appear promising enough to warrant continued research and development in a variety of areas including evaluation, integration with graphical interfaces, and application of other technologies for segmentation (also see section 1.7.2 for a discussion of spatial audio and auditory streaming techniques).

6.3 Evaluation of Segmentation Techniques

Automatically segmenting speech recordings based on features of conversational speech is a powerful and important step toward making it more efficient to listen to recorded speech. The techniques described in earlier chapters are successful at extracting information from recordings. Along with informal evaluations, such as those described in sections 5.12 and 6.1, it is necessary to develop more formalized measurement methods to extend and refine these speech processing techniques.

Part of the problem of evaluation is in precisely defining the information that one wants to extract from the speech signal. Finding the “major points” in a speech recording is a subjective measure based on high-level semantic and pragmatic information in the mind of the listener. Creating software that can automatically locate acoustic correlates of these features is difficult.

Automatically locating “emphasized” or “stressed” (O’Shaughnessy 1987) portions of a recording is easier, but emphasis is not always correlated with major topics. A talker, for example, may use emphasis for humor rather than as an indication of a new or important point. Some talkers also tend to emphasize just about everything they say, making it hard to identify important segments.

Perhaps the best way to evaluate such a system is to have a large database of appropriately labeled speech data. This labeling is a time consuming manual process. A variety of speech databases are available,[74] but much of the existing labeling has been oriented toward

[74]Such as through the Linguistic Data Consortium at the University of Pennsylvania.

speech recognition systems rather than high-level information based on the prosody of spontaneous speech.

In a study of automatically detecting emphasis and creating summaries (Chen 1992) several methods were used to obtain time-aligned emphasis labels. Subjects listened to speech recordings (in real time) and rated regions of the recordings on three levels of emphasis; other subjects listened to the same material and selected portions to create a summary. In another portion of the work, subjects identified emphasized words from short (2–5 second) phrases extracted from telephone conversations.[75] The hand-labeled summary data was used to develop an emphasis detection system and to evaluate the summaries that were created automatically.

Another method of evaluating the segments selected from a recording is to have subjects compare the results of different segmentation techniques. In the usability test (section 5.10) this type of evaluation was used to rate pitch-based versus isochronous segmentation methods. This style of comparison is useful for obtaining relative measures of perceived effectiveness. Note that in this portion of the test the touchpad interface was not used; subjects rated only the segmentation, not the interactive user interface for accessing the segments.

6.3.1 Combining SpeechSkimmer with a Graphical Interface

While this research has focused on non-visual interfaces, the techniques developed can be combined with graphical or other visual interfaces.

A visual component could be added to SpeechSkimmer in a variety of ways. The most basic change would be to make the skimming template active, so there is a graphical indication of the current speed, skimming level, and location within the recording (i.e., a time line display). The system could also be integrated into a full workstation-based graphical user interface. Besides mapping the fundamental SpeechSkimmer controls to a mouse-based system, it is possible to add a variety of visual cues, such as a real-time version of figure 5-5, to aid in the skimming process. Note that one must be careful not to overload the visual system since the user’s eye may be busy (e.g., watching video images).

Existing graphical interfaces for manipulating temporal media that contain speech can be enhanced with SpeechSkimmer technology. For example, the video streamer (Elliott 1993) and Media Streams (Davis 1993) systems make primary use of the visual channel for annotating,

---

[75]Subjects were not required to perform this task in real time.

logging, editing, and visualizing the structure of video. These kinds of systems have concentrated on visual tools and techniques for navigating in video, and could be enhanced by adding the speech skimming techniques explored in this dissertation.

These video-based interfaces can be tied to the speech skimming interface and vice versa. For example, when quickly flipping through a set of video images, only the related high-level segments of speech could be played, rather than playing the random snippets of audio associated with the displayed frames. Similarly, the SpeechSkimmer interface (or perhaps a mouse-based version of it) can be used to skim through the audio track while the related video images are synchronously displayed.

6.3.2 Segmentation by Speaker Identification

Acoustically based speaker identification can provide a powerful cue for segmentation and information retrieval in speech systems. For example, when searching for a piece of information within a recording, the search space can be greatly reduced if individual talkers can be identified (e.g., “play only things Marc said”).

The SpeechSkimmer system has been used with speaker identification-based segmentation. A two person conversation was analyzed with speaker identification software (Reynolds 1993) that determined when each talker was active. These data were translated into SpeechSkimmer format such that level 1 represented the entire conversation; jumping took the listener to the next turn change in the conversation. Level 2 played only the speech from one talker, while level 3 played the speech from the other. Jumping within these levels brought the listener to start of that talker’s next conversational turn.

6.3.3 Segmentation by Word Spotting

Keyword spotting can also be used for segmentation, and incorporated into the speech skimming system. Keywords found in recorded utterances can be used as text tags to allow for flexible information retrieval. Higher-level summarization or content-based retrieval methods, however, such as the gisting techniques described in section 1.4.2, will ultimately prove more useful. Such gisting systems may become common as recognition technology continues to evolve, but may be most useful for information access when combined with the skimming ideas presented here.

6.4 Summary

Speech is naturally slow to listen to, and difficult to skim. This research attempts to transcend these limitations, making it easier and more efficient to consume recorded speech through interaction and processing techniques. By combining segmentation techniques that extract structural information from spontaneous speech with a hierarchical representation and an interactive listener control, it is possible to overcome the time bottleneck in speech-based systems. The systems presented here provide “intelligent” filtering of recorded speech; the intelligence is provided by the interactive control of the human, in combination with the speech segmentation techniques.

An effort has been made to present this research clearly and simply. Many of the techniques and systems described herein may seem obvious in retrospect, but these solutions were untried and unknown when this research began. Initial system prototypes were more complex, and therefore more difficult to use and describe. In simplicity there is elegance.

The Hyperspeech exploration system was compelling to use, or listen to, for many reasons. First, interacting with a computer by speech is very powerful, particularly when the same modality is used for both input and output. Second, speech is a very rich communications medium, layers of meaning can be embedded in intonation that cannot be adequately captured by text alone. Third, listening to speech is “easier” than reading text—it takes less effort to listen to a lecture than to read a paper on the same subject. Finally, it is not desirable, or necessary, to look at a screen during an interaction. The bulk of the Hyperspeech user interface was debugged by conversing with the system while wandering around the room and looking out the window. In speech-only systems, the hands, eyes, and body are free.

Just as it is necessary to go beyond the “keyword barrier” to partially understanding text in advanced information retrieval systems (Mauldin 1989), we must go beyond the “time compression barrier” to understand the content of speech recordings in new audio retrieval systems. SpeechSkimmer is an important advance in this direction through the synergy of segmentation and interface techniques. When asked if the system was useful, one test subject commented “Yes, definitely. It’s quite nice, I would use it to listen to talks or lectures that I missed … it would be super, I would do it all the time. I don’t do it now since it would require me to sit through the duration of the two hour [presentations] …”

This dissertation presents a framework for thinking about and designing speech skimming systems. The fundamental mechanisms presented here allow other types of segmentation or new interface techniques to be easily plugged in. Note also that SpeechSkimmer is not only intended to be an application in itself, but rather a technology to be incorporated into any interface that uses recorded speech. Skimming techniques, such as those developed here, enable speech to be readily accessed in a range of applications and devices, empowering a new generation of user interfaces that use speech. When discussing the SpeechSkimmer system, one of the usability test subjects put it cogently: “it is a concept, not a box.”

This research provides insight into making one’s ears an alternative to one’s eyes as a means for accessing stored information. Tufte said “Unlike speech, visual displays are simultaneously a wideband and a perceiver-controllable channel” (Tufte 1990, 31). This dissertation attempts to overcome these conventional notions, increasing the information bandwidth of the speech channel and allowing the perceiver to interactively control access to speech information. Speech is a powerful medium, and its use in computer-based systems will expand in unforeseen ways as tools and techniques, such as those described here, allow a user to interactively skim, and efficiently listen to, recorded speech.

Glossary

cm	centimeter
CSCW	computer supported cooperative work
dB	decibel
dichotic	a different signal is presented to each ear
diotic	the same signal is presented to both ears
DSI	Digital Speech Interpolation
F0	fundamental frequency of voicing
HMM	Hidden Markov Model
Hz	frequency in Hertz (cycles per second)
isochronous	recurring at regular intervals
I/O	input/output
kg	kilogram
kHz	kilohertz
LPC	linear predictive coding
monotic	a signal is presented to only one ear
ms	milliseconds, 1/1000 of a second (e.g., 250 ms = 1/4 s)
pitch	see F0
RMS	root mean square
s	second
SNR	signal to noise ratio
SOLA	synchronized overlap add method of time compression
TASI	Time Assigned Speech Interpolation
wpm	words per minute
ZCR	zero crossing rate (crossings per second)
µs	microseconds, 1/1000000 of a second

References

Aaronson 1971 — D. Aaronson, N. Markowitz, and H. Shapiro. Perception and Immediate Recall of Normal and Compressed Auditory Sequences. *Perception and Psychophysics 9*, 4 (1971), 338–344.

Adaptive 1991 — Adaptive Digital Systems Inc. *JBIRD Specifications*, Irvine, CA. 1991.

Agnello 1974 — J. G. Agnello. Review of the Literature on the Studies of Pauses. In *Time-Compressed Speech*, edited by S. Duker. Scarecrow, 1974. pp. 566–572.

Arons 1989 — B. Arons, C. Binding, K. Lantz, and C. Schmandt. The VOX Audio Server. In *Proceedings of the 2nd IEEE ComSoc International Multimedia Communications Workshop,* IEEE Communications Society, Apr. 1989.

Arons 1991a — B. Arons. Hyperspeech: Navigating in Speech-Only Hypermedia. In *Proceedings of Hypertext (San Antonio, TX, Dec. 15–18),* ACM, New York, 1991, pp. 133–146.

Arons 1991b — B. Arons. Authoring and Transcription Tools for Speech-Based Hypermedia Systems. In *Proceedings of 1991 Conference,* American Voice I/O Society, Sep. 1991, pp. 15–20.

Arons 1992a — B. Arons. Techniques, Perception, and Applications of Time-Compressed Speech. In *Proceedings of 1992 Conference,* American Voice I/O Society, Sep. 1992, pp. 169–177.

Arons 1992b — B. Arons. A Review of the Cocktail Party Effect. *Journal of the American Voice I/O Society 12* (Jul. 1992), 35–50.

Arons 1992c — B. Arons. Tools for Building Asynchronous Servers to Support Speech and Audio Applications. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST),* ACM SIGGRAPH and ACM SIGCHI, ACM Press, Nov. 1992, pp. 71–78.

Arons 1993a — B. Arons. SpeechSkimmer: Interactively Skimming Recorded Speech. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST),* ACM SIGGRAPH and ACM SIGCHI, ACM Press, Nov. 1993, pp. 187–196.

Arons 1993b — B. Arons. Hyperspeech (videotape). *ACM SIGGRAPH Video Review 88* (1993). InterCHI ’93 Technical Video Program.

Atal 1976 — B. S. Atal and L. R. Rabiner. A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24*, 3 (Jun. 1976), 201–212.

Backer 1982 — D. S. Backer and S. Gano. Dynamically Alterable Videodisc Displays. In *Proceedings of Graphics Interface 82,* 1982.

Ballou 1987 — G. Ballou. *Handbook for Sound Engineers.* Indianapolis, IN: Howard W. Sams and Company, 1987.

Beasley 1976 — D. S. Beasley and J. E. Maki. Time- and Frequency-Altered Speech. Ch. 12 in *Contemporary Issues in Experimental Phonetics*, edited by N. J. Lass. New York: Academic Press, 1976. pp. 419–458.

Birkerts 1993 — S. Birkerts. Close Listening. *Harper's Magazine 286* (Jan. 1993), 86–91. Reprinted as Have You Heard the Word in Unte Reader, Jul./Aug. 1993, 110–111.

Blattner 1989 — M. M. Blattner, D. A. Sumikawa, and R. M. Greenberg. Earcons and Icons: Their Structure and Common Design Principles. *Human Computer Interaction 4*, 1 (1989), 11–44.

Bly 1982 — S. Bly. Presenting Information in Sound. In *Proceedings of the CHI '82 Conference on Human Factors in Computer Systems,* ACM, New York, 1982, pp. 371–375.

Brady 1965 — P. T. Brady. A Technique for Investigating On-Off Patterns of Speech. *The Bell System Technical Journal 44*, 1 (Jan. 1965), 1–22.

Brady 1968 — P. T. Brady. A Statistical Analysis of On-Off Patterns in 16 Conversations. *The Bell System Technical Journal 47*, 1 (Jan. 1968), 73–91.

Brady 1969 — P. T. Brady. A Model for Generating On-Off Speech Patterns in Two-Way Conversation. *The Bell System Technical Journal 48* (Sep. 1969), 2445–2472.

Bregman 1990 — A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, MA: MIT Press, 1990.

Bush 1945 — V. Bush. As We May Think. *Atlantic Monthly 176*, 1 (Jul. 1945), 101–108.

Butterworth 1977 — B. Butterworth, R. R. Hine, and K. D. Brady. Speech and Interaction in Sound-only Communication Channels. *Semiotica 20-1/2* (1977), 81–99.

Buxton 1991 — W. Buxton, B. Gaver, and S. Bly. *The Use of Non-Speech Audio at the Interface*. ACM SIGGCHI. Tutorial Notes. 1991.

Campanella 1976 — S. J. Campanella. Digital Speech Interpolation. *COMSAT Technical Review 6*, 1 (Spring 1976), 127–158.

Card 1991 — S. K. Card, J. D. Mackinlay, and G. G. Robertson. A Morphological Analysis of the Design Space of Input Devices. *ACM Transactions on Information Systems 9*, 2 (Apr. 1991), 99–122.

Chalfonte 1991 — B. L. Chalfonte, R. S. Fish, and R. E. Kraut. Expressive Richness: A Comparison of Speech and Text as Media for Revision. In *Proceedings of CHI (New Orleans, LA, Apr. 28–May 2),* ACM, New York, 1991, pp. 21–26.

Chen 1992 — F. R. Chen and M. Withgott. The Use of Emphasis to Automatically Summarize Spoken Discourse. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1992, pp. 229–233.

Cherry 1954 — E. C. Cherry and W. K. Taylor. Some Further Experiments on the Recognition of Speech, with One and Two Ears. *Journal of the Acoustic Society of America 26* (1954), 554–559.

Cohen 1991 — M. Cohen and L. F. Ludwig. Multidimensional Window Management. *International Journal of Man/Machine Studies 34* (1991), 319–336.

Cohen 1993 — M. Cohen. Integrating Graphic and Audio Windows. *Presence 1*, 4 (Fall 1993), 468–481.

Compernolle 1990 — D. van Compernolle, W. Ma, F. Xie, and M. van Diest. Speech Recognition in Noisy Environments with the Aid of Microphone Arrays. *Speech Communication 9* (1990), 433–442.

Condray 1987 — R. Condray. Speed Listening: Comprehension of Time-Compressed Telegraphic Speech. Ph.D. dissertation, University of Nevada-Reno, 1987.

Conklin 1987 — J. Conklin. Hypertext: an Introduction and Survey. *IEEE Computer 20*, 9 (Sep. 1987), 17–41.

Davenport 1991 — G. Davenport, T. A. Smith, and N. Pincever. Cinematic Primitives for Multimedia. *IEEE Computer Graphics and Applications* (Jul. 1991), 67–74.

David 1956 — E. E. David and H. S. McDonald. Note on Pitch-Synchronous Processing of Speech. *Journal of the Acoustic Society of America 28*, 7 (1956), 1261–1266.

Davis 1993 — M. Davis. Media Streams: An Iconic Visual Language for Video Annotation. In *IEEE/CS Symposium on Visual Languages,* Bergen, Norway: Aug. 1993.

de Souza 1983 — P. de Souza. A Statistical Approach to the Design of an Adaptive Self-Normalizing Silence Detector. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-31*, 3 (Jun. 1983), 678–684.

Degen 1992 — L. Degen, R. Mander, and G. Salomon. Working with Audio: Integrating Personal Tape Recorders and Desktop Computers. In *Proceedings of CHI (Monterey, CA, May 3–7),* ACM, New York, 1992, pp. 413–418.

Dolson 1986 — M. Dolson. The Phase Vocoder: A tutorial. *Computer Music Journal 10*, 4 (1986), 14–27.

Drago 1978 — P. G. Drago, A. M. Molinari, and F. C. Vagliani. Digital Dynamic Speech Detectors. *IEEE Transactions on Communications COM-26*, 1 (Jan. 1978), 140–145.

Duker 1974 — S. Duker. Summary of Research on Time-Compressed Speech. In *Time-Compressed Speech*, edited by S. Duker. Scarecrow, 1974. pp. 501–508.

Durlach 1992 — N. I. Durlach, A. Rigopulos, X. D. Pang, W. S. Woods, A. Kulkarni, H. S. Colburn, and E. M. Wenzel. On the Externalization of Auditory Images. *Presence 1*, 2 (1992), 251–257.

Edwards 1993 — A. D. N. Edwards and R. D. Stevens. Mathematical Representations: Graphs, Curves and Formulas. In *INSERM Seminar on Non-Visual Presentations of Data in Human-Computer Interactions,* Paris: Mar. 1993.

Elliott 1993	E. L. Elliott. Watch-Grab-Arrange-See: Thinking with Motion Images via Streams and Collages. Master's thesis, Media Arts and Sciences Section, MIT, 1993.
Engelbart 1984	D. Engelbart. Authorship provisions in AUGMENT. In *IEEE CompCon Proceedings,* Spring 1984, pp. 465–472.
Ericsson 1984	K. A. Ericsson and H. A. Simon. *Protocol Analysis: Verbal Reports as Data.* Cambridge, MA: MIT Press, 1984.
Fairbanks 1954	G. Fairbanks, W. L. Everitt, and R. P. Jaeger. Method for Time or Frequency Compression-Expansion of Speech. *Transactions of the Institute of Radio Engineers, Professional Group on Audio AU-2* (1954), 7–12. Reprinted in G. Fairbanks, Experimental Phonetics: Selected Articles, University of Illinois Press, 1966.
Fairbanks 1957	G. Fairbanks and F. Kodman. Word Intelligibility as a Function of Time Compression. *Journal of the Acoustic Society of America 29* (1957), 636–641. Reprinted in G. Fairbanks, Experimental Phonetics: Selected Articles, University of Illinois Press, 1966.
Flanagan 1985	J. L. Flanagan, J. D. Johnson, R. Zahn, and G. W. Elko. Computer-Steered Microphone Arrays for Sound Transduction in Large Rooms. *Journal of the Acoustic Society of America 78*, 5 (Nov. 1985), 1508–1518.
Foulke 1969	W. Foulke and T. G. Sticht. Review of Research on the Intelligibility and Comprehension of Accelerated Speech. *Psychological Bulletin 72* (1969), 50–62.
Foulke 1971	E. Foulke. The Perception of Time Compressed Speech. Ch. 4 in *Perception of Language*, edited by P. M. Kjeldergaard, D. L. Horton, and J. J. Jenkins. Charles E. Merrill Publishing Company, 1971. pp. 79–107.
Furnas 1986	G. W. Furnas. Generalized Fisheye Views. In *Proceedings of CHI (Boston, MA),* ACM, New York, 1986, pp. 16–23.
Gan 1988	C. K. Gan and R. W. Donaldson. Adaptive Silence Deletion for Speech Storage and Voice Mail Applications. *IEEE Transactions on Acoustics, Speech, and Signal Processing 36*, 6 (Jun. 1988), 924–927.
Garvey 1953a	W. D. Garvey. The Intelligibility of Abbreviated Speech Patterns. *Quarterly Journal of Speech 39* (1953), 296–306. Reprinted in J. S. Lim, editor, Speech Enhancement, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.
Garvey 1953b	W. D. Garvey. The Intelligibility of Speeded Speech. *Journal of Experimental Psychology 45* (1953), 102–108.
Gaver 1989a	W. W. Gaver. Auditory Icons: Using Sound in Computer Interfaces. *Human-Computer Interaction 2* (1989), 167–177.
Gaver 1989b	W. W. Gaver. The SonicFinder: An Interface that uses Auditory Icons. *Human-Computer Interaction 4*, 1 (1989), 67–94.
Gaver 1993	W. W. Gaver. Synthesizing Auditory Icons. In *Proceedings of INTERCHI (Amsterdam, The Netherlands, Apr. 24–29),* SIGGCHI, ACM, New York, 1993, pp. 228–235.

Gerber 1974 S. E. Gerber. Limits of Speech Time Compression. In *Time-Compressed Speech*, edited by S. Duker. Scarecrow, 1974. pp. 456–465.

Gerber 1977 S. E. Gerber and B. H. Wulfeck. The Limiting Effect of Discard Interval on Time-Compressed Speech. *Language and Speech 20*, 2 (1977), 108–115.

Glavitsch 1992 U. Glavitsch and P. Schäuble. A System for Retrieving Speech Documents. In *15th Annual International SIGIR '92,* ACM, New York, 1992, pp. 168–176.

Grice 1975 H. P. Grice. Logic and Conversation. In *Speech Acts*, edited by P. Cole and J. L. Morgan. Syntax and Semantics, vol. 3. New York: Academic Press, 1975. pp. 41–58.

Griffin 1984 D. W. Griffin and J. S. Lim. Signal Estimation from Modified Short-Time Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-32*, 2 (Apr. 1984), 236–243.

Gruber 1982 J. G. Gruber. A Comparison of Measured and Calculated Speech Temporal Parameters Relevant to Speech Activity Detection. *IEEE Transactions on Communications COM-30*, 4 (Apr. 1982), 728–738.

Gruber 1983 J. G. Gruber and N. H. Le. Performance Requirements for Integrated Voice/Data Networks. *IEEE Journal on Selected Areas in Communications SAC-1*, 6 (Dec. 1983), 981–1005.

Grudin 1988 J. Grudin. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. In *Proceedings of CSCW (Portland, OR, Sep. 26–28),* ACM, New York, 1988, pp. 85–93.

Hardam 1990 E. Hardam. High Quality Time-Scale Modification of Speech Signals Using Fast Synchronized-Overlap-Add Algorithms. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1990, pp. 409–412.

Hawley 1993 M. Hawley. Structure out of Sound. Ph.D. dissertation, MIT, Sep. 1993.

Hayes 1983 P. J. Hayes and D. R. Reddy. Steps Towards Graceful Interaction in Spoken and Written Man-machine Communication. *International Journal of Man/Machine Studies 19* (1983), 231–284.

Heiman 1986 G. W. Heiman, R. J. Leo, G. Leighbody, and K. Bowler. Word Intelligibility Decrements and the Comprehension of Time-Compressed Speech. *Perception and Psychophysics 40*, 6 (1986), 407–411.

Hejna 1990 D. J. Hejna Jr. Real-Time Time-Scale Modification of Speech via the Synchronized Overlap-Add Algorithm. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, Feb. 1990.

Hess 1976 W. J. Hess. A Pitch-Synchronous Digital Feature Extraction System for Phonemic Recognition of Speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24*, 1 (Feb. 1976), 14–25.

Hess 1983 W. Hess. *Pitch Determination of Speech Signals: Algorithms and Devices.* Berlin and New York: Springer-Verlag, 1983.

Hindus 1993 D. Hindus, C. Schmandt, and C. Horner. Capturing, Structuring, and Representing Ubiquitous Audio. *ACM Transactions on Information Systems 11*, 4 (Oct. 1993), 376–400.

Hirschberg 1986 J. Hirschberg and J. Pierrehumbert. The Intonational Structuring of Discourse. In *Proceedings of the Association for Computational Linguistics,* 1986, pp. 136–144.

Hirschberg 1987 J. Hirschberg and Diane Litman. Now Let's Talk About Now: Identifying Cue Phrases Intonationally. In *Proceedings of the Conference (Stanford, CA, Jul. 6–9),* Association for Computational Linguistics, 1987, pp. 163–171.

Hirschberg 1992 J. Hirschberg and B. Grosz. Intonational Features of Local and Global Discourse. In *Proceedings of the Speech and Natural Language workshop (Harriman, NY, Feb.23-26),* Defense Advanced Research Projects Agency, San Mateo, CA: Morgan Kaufmann Publishers, 1992, pp. 441–446.

Houle 1988 G. R. Houle, A. T. Maksymowicz, and H. M. Penafiel. Back-End Processing for Automatic Gisting Systems. In *Proceedings of 1988 Conference,* American Voice I/O Society, 1988.

Hu 1987 A. Hu. *Automatic Emphasis Detection in Fluent Speech with Transcription*, Unpublished MIT Bachelor's thesis, May, 1987.

Jankowski 1976 J. A. Jankowski. A New Digital Voice-Activated Switch. *COMSAT Technical Review 6*, 1 (Spring 1976), 159–178.

Jeffries 1991 R. Jeffries, J. R. Miller, C. Wharton, and K. M. Uyeda. User Interface Evaluation in the Real World: A Comparison of Four Techniques. In *Proceedings of CHI (New Orleans, LA, Apr. 28–May 2),* ACM, New York, Apr 1991, pp. 119–124.

Kato 1992 Y. Kato and K. Hosoya. Fast Message Searching Method for Voice Mail Service and Voice Bulletin Board Service. In *Proceedings of 1992 Conference,* American Voice I/O Society, 1992, pp. 215–222.

Kato 1993 Y. Kato and K. Hosoya. Message Browsing Facility for Voice Bulletin Board Service. In *Human Factors in Telecommunications '93,* 1993, pp. 321–330.

Keller 1992 E. Keller. *Signalyze: Signal Analysis for Speech and Sound User's Manual*. InfoSignal Inc., Lausanne, Switzerland. 1992.

Keller 1993 E. Keller. *Apple Microphone Inputs*, Technical notes distributed with the Signalyze software package for the Macintosh (76357.1213@compuserve.com), 1993.

Kernighan 1976 B. W. Kernighan and P. J. Plauger. *Software Tools.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1976.

Klatt 1987 D. H. Klatt. Review of Text-To-Speech Conversion for English. *Journal of the Acoustic Society of America 82* (Sep. 1987), 737–793.

Knuth 1984 D. E. Knuth. *The TEXbook.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1984.

Kobatake 1989 H. Kobatake, K. Tawa, and A. Ishida. Speech/Nonspeech Discrimination for Speech Recognition System, Under Real Life Noise Environments. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1989, pp. 365–368.

Lamel 1981 — L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon. An Improved Endpoint Detector for Isolated Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-29*, 4 (Aug. 1981), 777–785.

Lamming 1991 — M. G. Lamming. Towards a Human Memory Prosthesis. Xerox EuroPARC, technical report no. EPC-91-116 1991.

Lamport 1986 — L. Lamport. *LATEX: A Document Preparation System.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1986.

Lass 1977 — N. J. Lass and H. A. Leeper. Listening Rate Preference: Comparison of Two Time Alteration Techniques. *Perceptual and Motor Skills 44* (1977), 1163–1168.

Lee 1972 — F. F. Lee. Time Compression and Expansion of Speech by the Sampling Method. *Journal of the Audio Engineering Society 20*, 9 (Nov. 1972), 738–742.

Lee 1986 — H. H. Lee and C. K. Un. A Study of On-off Characteristics of Conversational Speech. *IEEE Transactions on Communications COM-34*, 6 (Jun. 1986), 630–637.

Levelt 1989 — W. J. M. Levelt. *Speaking: From Intention to Articulation.* Cambridge, MA: MIT Press, 1989.

Lim 1983 — J. S. Lim. *Speech Enhancement.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

Lipscomb 1993 — J. S. Lipscomb and M. E. Pique. Analog Input Device Physical Characteristics. *SIGCHI Bulletin 25*, 3 (Jul. 1993), 40–45.

Ludwig 1990 — L. Ludwig, N. Pincever, and M. Cohen. Extending the Notion of a Window System to Audio. *IEEE Computer 23*, 8 (Aug. 1990), 66–72.

Lynch 1987 — J. F. Lynch Jr., J. G. Josenhans, and R. E. Crochiere. Speech/Silence Segmentation for Real-Time Coding via Rule Based Adaptive Endpoint Detection. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1987, pp. 1348–1351.

Mackinlay 1991 — J. D. Mackinlay, G. G. Robertson, and S. K. Card. The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of CHI (New Orleans, LA, Apr. 28–May 2),* ACM, New York, 1991, pp. 173–179.

Makhoul 1986 — J. Makhoul and A. El-Jaroudi. Time-Scale Modification in Medium to Low Rate Coding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1986, pp. 1705–1708.

Maksymowicz 1990 — A. T. Maksymowicz. Automatic Gisting for Voice Communications. In *IEEE Aerospace Applications Conference,* IEEE, Feb. 1990, pp. 103–115.

Malah 1979 — D. Malah. Time-Domain Algorithms for Harmonic Bandwidth Reduction and Time Scaling of Speech Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-27*, 2 (Apr. 1979), 121–133.

Malone 1988 — T. W. Malone, K. R. Grant, K-Y. Lai, R. Rao, and D. Rosenblitt. Semi-Structured Messages are Surprisingly Useful for Computer-Supported Coordination. In *Computer-Supported Cooperative Work: A Book of Readings*, edited by I. Greif. Morgan Kaufmann Publishers, 1988.

Manandhar 1991 S. Manandhar. Activity Server: You Can Run but you Can't Hide. In *Proceedings of the Summer 1991 USENIX Conference,* Usenix, 1991.

Mauldin 1989 M. L. Mauldin. Information Retrieval by Text Skimming. Carnegie Mellon University Ph.D. dissertation, School of Computer Science, technical report no. CMU-CS-89-193, Aug. 1989.

Maxemchuk 1980 N. Maxemchuk. An Experimental Speech Storage and Editing Facility. *Bell System Technical Journal 59*, 8 (Oct. 1980), 1383–1395.

McAulay 1986 R. J. McAulay and T. F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-34* (Aug. 1986), 744–754.

Mermelstein 1975 P. Mermelstein. Automatic Segmentation of Speech into Syllabic Units. *Journal of the Acoustic Society of America 58*, 4 (Oct. 1975), 880–883.

Microtouch 1992 Microtouch Systems Inc. *UnMouse User's Manual*, Wilmington, MA. 1992.

Miedema 1962 H. Miedema and M. G. Schachtman. TASI Quality—Effect of Speech Detectors and Interpolators. *The Bell System Technical Journal* (1962), 1455–1473.

Miller 1950 G. A. Miller and J. C. R. Licklider. The Intelligibility of Interrupted Speech. *Journal of the Acoustic Society of America 22*, 2 (1950), 167–173.

Mills 1992 M. Mills, J. Cohen, and Y. Y. Wong. A Magnifier Tool for Video Data. In *Proceedings of CHI (Monterey, CA, May 3–7),* ACM, New York, Apr. 1992, pp. 93–98.

Minifie 1974 F. D. Minifie. Durational Aspects of Connected Speech Samples. In *Time-Compressed Speech*, edited by S. Duker. Scarecrow, 1974. pp. 709–715.

Muller 1990 M. J. Muller and J. E. Daniel. Toward a Definition of Voice Documents. In *Conference on Office Information Systems (Cambridge, MA, Apr. 25–27),* ACM SIGOIS and IEEECS TC-OA, ACM Press, 1990, pp. 174–183. SIGOIS Bulletin Vol. 11, Issues 2–3

Mullins 1993 A. Mullins. *Hypernews: Organizing Audio News for Interactive Presentation*, Speech Research Group Technical Report, Media Laboratory, 1993.

Multimedia 1989 Multimedia Lab. *The Visual Almanac: An Interactive Multimedia Kit*, Apple Multimedia Lab, San Francisco, 1989.

Natural 1988 Natural MicroSystems Corporation. *ME/2 Device Driver Reference Manual*. Natick, MA, 1988.

Negroponte 1991 N. Negroponte. Beyond the Desktop Metaphor. Ch. 9 in *Research Directions in Computer Science: An MIT Perspective*, edited by A. R. Meyer, J. V. Guttag, R. L. Rivest, and P. Szolovits. Cambridge, MA: MIT Press, 1991. pp. 183–190.

Nelson 1974 T. Nelson. *Computer Lib: You Can and Must Understand Computers Now.* Hugo's Book Service, 1974.

Neuburg 1978 E. P. Neuburg. Simple Pitch-Dependent Algorithm for High Quality Speech Rate Changing. *Journal of the Acoustic Society of America 63*, 2 (1978), 624–625.

Nielsen 1990 J. Nielsen and R. Molich. Heuristic Evaluation of User Interfaces. In *Proceedings of CHI (Seattle, WA, Apr. 1–5),* ACM, New York, 1990.

Nielsen 1991 J. Nielsen. Finding Usability Problems through Heuristic Evaluation. In *Proceedings of CHI (New Orleans, LA, Apr. 28–May 2),* ACM, New York, Apr. 1991, pp. 373–380.

Nielsen 1993a J. Nielsen. *Usability Engineering.* San Diego: Academic Press, 1993.

Nielsen 1993b J. Nielsen. Is Usability Engineering Really Worth It? *IEEE Software 10*, 6 (Nov. 1993), 90–92.

Noll 1993 P. Noll. Wideband Speech and Audio Coding. *IEEE Communications Magazine 31*, 11 (Nov. 1993), 34–44.

Orr 1965 D. B. Orr, H. L. Friedman, and J. C. Williams. Trainability of Listening Comprehension of Speeded Discourse. *Journal of Educational Psychology 56* (1965), 148–156.

Orr 1971 D. B. Orr. A Perspective on the Perception of Time Compressed Speech. In *Perception of Language*, edited by P. M. Kjeldergaard, D. L. Horton, and J. J. Jenkins. Charles E. Merrill Publishing Company, 1971. pp. 108–119.

O'Shaughnessy 1987 D. O'Shaughnessy. *Speech Communication: Human and Machine.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1987.

O'Shaughnessy 1992 D. O'Shaughnessy. Recognition of Hesitations in Spontaneous Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1992, pp. I521–I524.

Parunak 1989 H. V. D. Parunak. Hypermedia Topologies and User Navigation. In *Proceedings of Hypertext (Pittsburgh, PA, Nov. 5–8),* ACM, New York, 1989, pp. 43–50.

Pincever 1991 N. C. Pincever. If You Could See What I Hear: Editing Assistance Through Cinematic Parsing. Master's thesis, Media Arts and Sciences Section, MIT, Jun. 1991.

Pitman 1985 K. M. Pitman. CREF: An Editing Facility for Managing Structured Text. MIT A. I. Memo, technical report no. 829, Feb. 1985.

Portnoff 1978 M. R. Portnoff. Time-Scale Modification of Speech Based on Short-Time Fourier Analysis. Ph.D. dissertation, MIT, Apr. 1978.

Portnoff 1981 M. R. Portnoff. Time-Scale Modification of Speech Based on Short-Time Fourier Analysis. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-29*, 3 (Jun. 1981), 374–390.

Quatieri 1986 T. F. Quatieri and R. J. McAulay. Speech Transformations Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-34* (Dec. 1986), 1449–1464.

Quereshi 1974 S. U. H. Quereshi. Speech Compression by Computer. In *Time-Compressed Speech*, edited by S. Duker. Scarecrow, 1974. pp. 618–623.

Rabiner 1975 L. R. Rabiner and M. R. Sambur. An Algorithm for Determining the Endpoints of Isolated Utterances. *The Bell System Technical Journal 54*, 2 (Feb. 1975), 297–315.

Rabiner 1989 L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE 77*, 2 (Feb. 1989), 257–286.

Raman 1992a T. V. Raman. An Audio View of (LA)TEX Documents. In *Proceedings of 13th Meeting TEX Users Group,* Portland, OR: Jul. 1992, pp. 372–379.

Raman 1992b T. V. Raman. Documents are not Just for Printing. In *Proceedings Principles of Document Processing,* Washington, DC: Oct. 1992.

Reich 1980 S. S. Reich. Significance of Pauses for Speech Perception. *Journal of Psycholinguistic Research 9*, 4 (1980), 379–389.

Resnick 1992a P. Resnick and R. A. Virzi. Skip and Scan: Cleaning Up Telephone Interfaces. In *Proceedings of CHI (Monterey, CA, May 3–7),* ACM, New York, Apr. 1992, pp. 419–426.

Resnick 1992b P. Resnick. HyperVoice: Groupware by Telephone. Ph.D. dissertation, MIT, 1992.

Resnick 1992c P. Resnick. HyperVoice a Phone-Based CSCW Platform. In *Proceedings of CSCW (Toronto, Ont., Oct. 31–Nov. 4),* SIGCHI and SIGOIS, ACM Press, 1992, pp. 218–225.

Reynolds 1993 D. A. Reynolds. A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification. Lincoln Laboratory, MIT, technical report no. 967, Lexington, MA, Feb. 1993.

Richaume 1988 A. Richaume, F. Steenkeste, P. Lecocq, and Y. Moschetto. Intelligibility and Comprehension of French Normal, Accelerated, and Compressed Speech. In *IEEE Engineering in Medicine and Biology Society 10th Annual International Conference,* 1988, pp. 1531–1532.

Rippey 1975 R. F. Rippey. Speech Compressors for Lecture Review. *Educational Technology* (Nov. 1975), 58–59.

Roe 1993 D. B. Roe and J. G. Wilpon. Whither Speech Recognition: The Next 25 Years. *IEEE Communications Magazine 31*, 11 (Nov. 1993), 54–62.

Rose 1991 R. C. Rose. Techniques for Information Retrieval from Speech Messages. *The Lincoln Lab Journal 4*, 1 (1991), 45–60.

Roucos 1985 S. Roucos and A. M. Wilgus. High Quality Time-Scale Modification for Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1985, pp. 493–496.

Salthouse 1984 T. A. Salthouse. The Skill of Typing. *Scientific American* (Feb. 1984), 128–135.

Savoji 1989 M. H. Savoji. A Robust Algorithm for Accurate Endpointing of Speech Signals. *Speech Communication 8* (1989), 45–60.

Schmandt 1984 C. Schmandt and B. Arons. A Conversational Telephone Messaging System. *IEEE Transactions on Consumer Electronics CE-30*, 3 (Aug. 1984), xxi–xxiv.

Schmandt 1985 — C. Schmandt, B. Arons, and C. Simmons. Voice Interaction in an Integrated Office and Telecommunications Environment. In *Proceedings of 1985 Conference,* American Voice I/O Society, 1985.

Schmandt 1986 — C. Schmandt and B. Arons. A Robust Parser and Dialog Generator for a Conversational Office System. In *Proceedings of 1986 Conference,* American Voice I/O Society, 1986, pp. 355–365.

Schmandt 1987 — C. Schmandt and B. Arons. Conversational Desktop (videotape). *ACM SIGGRAPH Video Review 27* (1987).

Schmandt 1988 — C. Schmandt and M. McKenna. An Audio and Telephone Server for Multi-Media Workstations. In *Proceedings of the 2nd IEEE Conference on Computer Workstations,* IEEE Computer Society, Mar. 1988, pp. 150–160.

Schmandt 1989 — C. Schmandt and B. Arons. Getting the Word (Desktop Audio). *Unix Review 7*, 10 (Oct. 1989), 54–62.

Schmandt 1993 — C. Schmandt. From Desktop Audio to Mobile Access: Opportunities for Voice in Computing. Ch. 8 in *Advances in Human-Computer Interaction*, edited by H. R. Hartson and D. Hix. Ablex Publishing Corporation, 1993. pp. 251–283.

Scott 1967 — R. J. Scott. Time Adjustment in Speech Synthesis. *Journal of the Acoustic Society of America 41*, 1 (1967), 60–65.

Scott 1972 — R. J. Scott and S. E. Gerber. Pitch-Synchronous Time-Compression of Speech. In *Conference on Speech Communication and Processing,* IEEE, 1972, pp. 63–65. Reprinted in J. S. Lim, editor, Speech Enhancement, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

Sheridan 1992a — T. B. Sheridan. Defining Our Terms. *Presence 1*, 2 (1992), 272–274.

Sheridan 1992b — T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control.* Cambridge, MA: MIT Press, 1992.

Silverman 1987 — K. E. A. Silverman. The Structure and Processing of Fundamental Frequency Contours. Ph.D. dissertation, University of Cambridge, Apr. 1987.

Smith 1970 — S. L. Smith and N. C. Goodwin. Computer-Generated Speech and Man-Computer Interaction. *Human Factors 12*, 2 (1970), 215–223.

Sony 1993 — Sony Corporation. *Telephone Answering Machine TAM-1000*. Document number 3-756-903-21(1). 1993.

Stallman 1979 — R. M. Stallman. EMACS: The Extensible, Customizable, Self-Documenting Display Editor. MIT A. I. Memo, technical report no. 519A. Revised Mar. 1981, Jun. 1979.

Stevens 1993 — R. D. Stevens. *Principles for Designing Systems for the Reading of Structured Information by Visually Disabled People*, Dissertation proposal, The Human Computer Interaction Group, Department of Computer Science, The University or York, 1993.

Sticht 1969 — T. G. Sticht. Comprehension of Repeated Time-Compressed Recordings. *The Journal of Experimental Education 37*, 4 (Summer 1969).

Stifelman 1991 L. J. Stifelman. Not Just Another Voice Mail System. In *Proceedings of 1991 Conference,* American Voice I/O Society, 1991, pp. 21–26.

Stifelman 1992a L. J. Stifelman. VoiceNotes: An Application for a Voice Controlled Hand-Held Computer. Master's thesis, Media Arts and Sciences Section, MIT, May 1992.

Stifelman 1992b L. Stifelman. *A Study of Rate Discrimination of Time-Compressed Speech*, Speech Research Group Technical Report, Media Laboratory, 1992.

Stifelman 1993 L. J. Stifelman, B. Arons, C. Schmandt, and E. A. Hulteen. VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker. In *Proceedings of INTERCHI (Amsterdam, The Netherlands, Apr. 24–29),* ACM, New York, 1993, pp. 179–186.

Thomas 1990 G. S. Thomas. *Xsim 2.0 Configurer's Guide*. Xsim, a general purpose tool for manipulating directed graphs, particularly Petri nets, is available from cs.washington.edu by anonymous ftp. 1990.

Toong 1974 H. D. Toong. A Study of Time-Compressed Speech. Ph.D. dissertation, MIT, Jun. 1974.

Tucker 1991 P. Tucker and D. M. Jones. Voice as Interface: An Overview. *International Journal of Human-Computer Interaction 3*, 2 (1991), 145–170.

Tufte 1990 E. Tufte. *Envisioning Information.* Cheshire, CT: Graphics Press, 1990.

Voor 1965 J. B. Voor and J. M. Miller. The Effect of Practice Upon the Comprehension of Time-Compressed Speech. *Speech Monographs 32* (1965), 452–455.

Wallace 1983 W. P. Wallace. Speed Listening: Exploring an Analogue of Speed Reading. University of Nevada- Reno, technical report no. NIE-G-81-0112, Feb. 1983.

Want 1992 R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems 10*, 1 (Jan. 1992), 91–102.

Watanabe 1990 T. Watanabe. The Adaptation of Machine Conversational Speed to Speaker Utterance Speed in Human-Machine Communication. *IEEE Transactions on Systems, Man, and Cybernetics 20*, 1 (1990), 502–507.

Watanabe 1992 T. Watanabe and T. Kimura. In Review of Acoustical Patents: #4,984,275 Method and Apparatus for Speech Recognition. *Journal of the Acoustic Society of America 92*, 4 (Oct. 1992), 2284.

Wayman 1988 J. L. Wayman and D. L. Wilson. Some Improvements on the Synchronized-Overlap-Add Method of Time-Scale Modification for Use in Real-Time Speech Compression and Noise Filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing 36*, 1 (Jan. 1988), 139–140.

Wayman 1989 J. L. Wayman, R. E. Reinke, and D. L. Wilson. High Quality Speech Expansion, Compression, and Noise Filtering Using the SOLA Method of Time Scale Modification. In *23d Asilomar Conference on Signals, Systems, and Computers,* vol. 2, Oct. 1989, pp. 714–717.

Webster 1971 — Webster. *Seventh New Collegiate Dictionary.* Springfield, MA: G. and C. Merriam Company, 1971.

Weiser 1991 — M. Weiser. The Computer for the 21st Century. *Scientific American 265*, 3 (Sep. 1991), 94–104.

Wenzel 1988 — E. M. Wenzel, F. L. Wightman, and S. H. Foster. A Virtual Display System for Conveying Three-Dimensional Acoustic Information. In *Proceedings of the Human Factors Society 32nd Annual Meeting,* 1988, pp. 86–90.

Wenzel 1992 — E. M. Wenzel. Localization in Virtual Acoustic Displays. *Presence 1*, 1 (1992), 80–107.

Wightman 1992 — C. W. Wightman and M. Ostendorf. Automatic Recognition of Intonational Features. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* vol. I, IEEE, 1992, pp. I221–I224.

Wilcox 1991 — L. Wilcox and M. Bush. HMM-Based Wordspotting for Voice Editing and Indexing. In *Eurospeech '91,* 1991, pp. 25–28.

Wilcox 1992a — L. Wilcox and M. Bush. Training and Search Algorithms for an Interactive Wordspotting System. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* IEEE, 1992.

Wilcox 1992b — L. Wilcox, I. Smith, and M. Bush. Wordspotting for Voice Editing and Audio Indexing. In *Proceedings of CHI (Monterey, CA, May 3–7),* ACM, New York, 1992, pp. 655–656.

Wilpon 1984 — J. G. Wilpon, L. R. Rabiner, and T. Martin. An Improved Word-Detection Algorithm for Telephone-Quality Speech Incorporating Both Syntactic and Semantic Constraints. *AT&T Bell Laboratories Technical Journal 63*, 3 (Mar. 1984), 479–497.

Wilpon 1990 — J. G. Wilpon, L. R. Rabiner, C. Lee, and E. R. Goldman. Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing 38*, 11 (Nov. 1990), 1870–1878.

Wingfield 1980 — A. Wingfield and K. A. Nolan. Spontaneous Segmentation in Normal and Time-compressed Speech. *Perception and Psychophysics 28*, 2 (1980), 97–102.

Wingfield 1984 — A. Wingfield, L. Lombardi, and S. Sokol. Prosodic Features and the Intelligibility of Accelerated Speech: Syntactic versus Periodic Segmentation. *Journal of Speech and Hearing Research 27* (Mar. 1984), 128–134.

Wolf 1992 — C. G. Wolf and J. R. Rhyne. Facilitating Review of Meeting Information Using Temporal Histories, IBM T. J. Watson Research Center, Working Paper 9/17. 1992.

Yatsuzuka 1982 — Y. Yatsuzuka. Highly Sensitive Speech Detector and High-Speed Voiceband Data Discriminator in DSI-ADPCM Systems. *IEEE Transactions on Communications COM-30*, 4 (Apr. 1982), 739–750.

Zellweger 1989 — P. T. Zellweger. Scripted Documents: A Hypermedia Path Mechanism. In *Proceedings of Hypertext (Pittsburgh, PA, Nov. 5–8),* ACM, New York, 1989, pp. 1–14.

Zemlin 1968 W. R. Zemlin, R. G. Daniloff, and T. H. Shriner. The Difficulty of Listening to Time-Compressed Speech. *Journal of Speech and Hearing Research 11* (1968), 875–881.