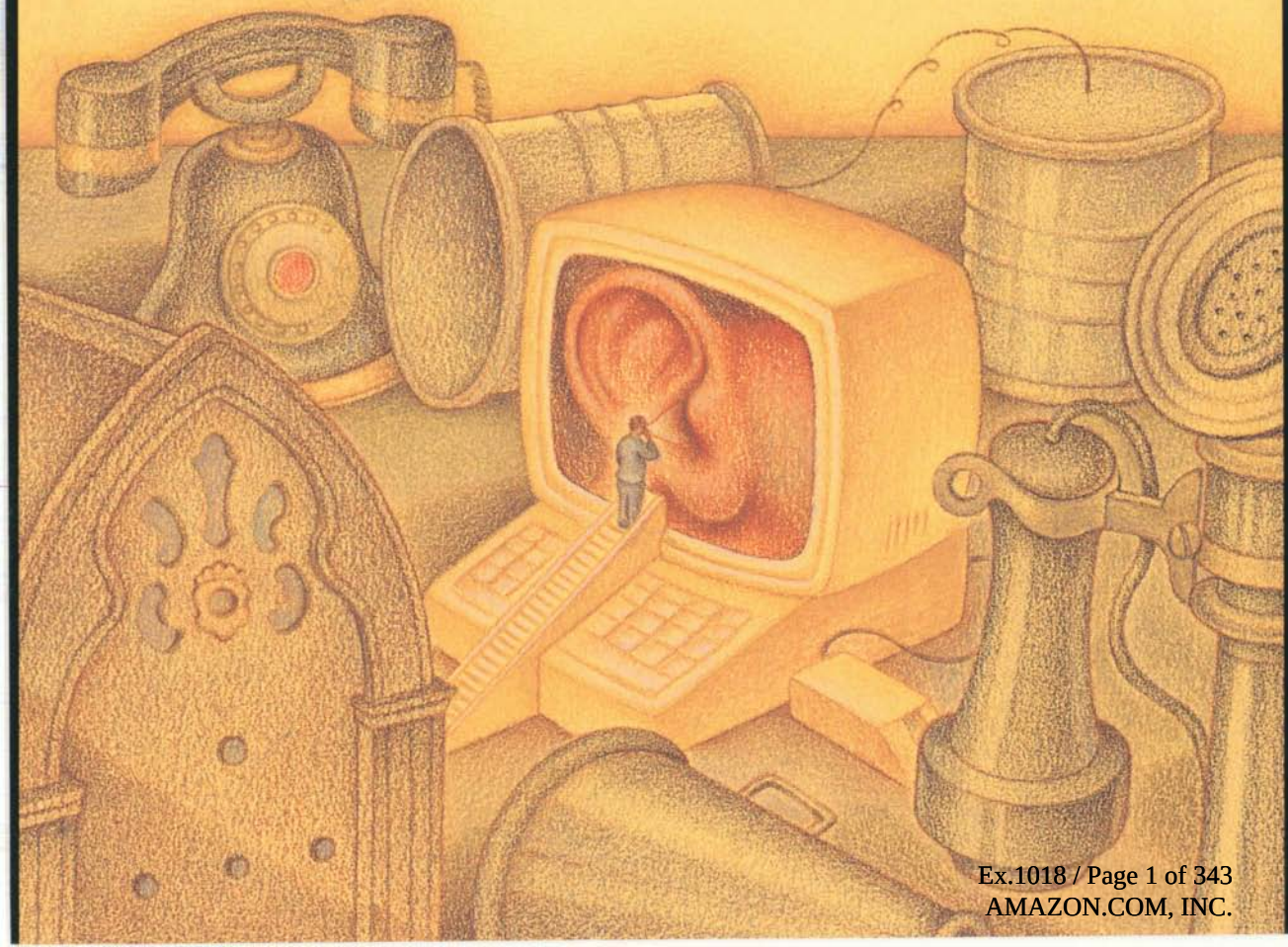
Christopher Schmandt
Foreword by Nicholas Negroponte
MIT Media Lab

Voice Communication with Computers

Conversational Systems



Ex.1018 / Page 1 of 343
AMAZON.COM, INC.

Voice Communication with Computers

Conversational Systems

Christopher Schmandt

This book discusses how human language techniques can be embedded in human-computer dialogue to make "talking computers" conversational. It uses an interdisciplinary approach to explain application of speech technologies to computer interfaces.

Chapters progress from physiological and psychological components of speech, language, and hearing to technical principles of speech synthesis and voice recognition to user interaction techniques to employ such technologies effectively. Areas covered include:

- Computational means of expressing linguistic and discourse knowledge
- Software architectures that support voice in multimedia computer environments
- The operations of digital recording, speech synthesis, and speech recognition
- Coding schemes based on data rate, intelligibility, and flexibility
- Applications and editing of stored voice in multimedia computer documents
- Integration of telephone functionality into computer workstations

Case studies demonstrate how to relate utterances to intention and real-world objects, use discourse knowledge to carry on the thread of a conversation across multiple exchanges, and integrate speech into many aspects of computer related work. Other topics addressed are:

• Desktop audio • Interactive voice response • ISDN—digital telephony • Text-to-speech algorithms • Interaction between audio and window systems • Operating system support and run-time support for voice • Applications that provide a means of accessing personal databases while away from the office



Interactive speech systems represent some of the newest technology available from several computer manufacturers. *Voice Communication with Computers* will provide vital information to application developers, user interface designers or researchers, human factors developers, groupware developers, and all professionals interested in taking advantage of the ability to speak with a computer.

About the Author

Christopher Schmandt is Director of the Speech Research Group, and a Principal Research Scientist at the Massachusetts Institute of Technology Media Laboratory.

Cover illustrations by Thomas Sciacca
Cover design by Angelo Papadopoulos

VAN NOSTRAND REINHOLD
115 Fifth Avenue, New York, NY 10003

ISBN 0-442-23935-1

Ex.1018 / Page 2 of 343
AMAZON.COM, INC.

VOICE COMMUNICATION WITH COMPUTERS

Conversational Systems

CHRISTOPHER SCHMANDT



VAN NOSTRAND REINHOLD
New York

Ex.1018 / Page 3 of 343
AMAZON.COM, INC.

Copyright © 1994 by Christopher Schmandt

Library of Congress Catalog Card Number 93-36404
ISBN 0-442-23935-1

All rights reserved. No part of this work covered by the copyright hereon may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the written permission of the publisher.

I(T)P Van Nostrand Reinhold is an International Thomson Publishing company. ITP logo is a trademark under license.

Printed in the United States of America

Van Nostrand Reinhold
115 Fifth Avenue
New York, NY 10003

International Thomson Publishing GmbH
Königswinterer Str. 418
53227 Bonn
Germany

International Thomson Publishing
Berkshire House,168-173
High Holborn, London WC1V 7AA
England

International Thomson Publishing Asia
221 Henderson Bldg. #05-10
Singapore 0315

Thomas Nelson Australia
102 Dodds Street
South Melbourne 3205
Victoria, Australia

International Thomson Publishing Japan
Kyowa Building, 3F
2-2-1 Hirakawacho
Chiyoda-Ku, Tokyo 102
Japan

Nelson Canada
1120 Birchmount Road
Scarborough, Ontario
M1K 5G4, Canada

ARCFF 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2

Library of Congress Cataloging-in-Publication Data
Schmandt, Chris.
Voice communication with computers / Chris Schmandt.
p. cm.
Includes bibliographical references and index.
ISBN 0-442-23935-1
1. Interactive computer systems. 2. Natural language processing (Computer science) I. Title.
QA76.9.I58S35 1993
006.4'54—dc20 93-36404
CIP

To Ava and Kaya for the patience to see me through writing this and helping me learn to hear the voice of the desert.

Contents

Speaking of Talk xvii
Preface xvii
Acknowledgments xxi
Introduction 1

Chapter 1. Speech as Communication 5

SPEECH AS CONVERSATION 6
HIERARCHICAL STRUCTURE OF CONVERSATION 8
REPRESENTATIONS OF SPEECH 12
Acoustic Representations 12
PHONEMES AND SYLLABLES 14
Phonemes 14
Syllables 17
Other Representations 17
SUMMARY 18

Chapter 2. Speech Production and Perception 19

VOCAL TRACT 19

Ex.1018 / Page 7 of 343
AMAZON.COM, INC.

THE SPEECH SOUNDS 24
Vowels 25
Consonants 26
Liquids and Glides 28
Acoustic Features of Phonemes 28
HEARING 28
Auditory System 29
Localization of Sounds 31
Psychoacoustics 33
SUMMARY 34
FURTHER READING 35

Chapter 3. Speech Coding 36

SAMPLING AND QUANTIZATION 37
SPEECH-CODING ALGORITHMS 44
Waveform Coders 44
Source Coders 51
CODER CONSIDERATIONS 53
Intelligibility 54
Editing 54
Silence Removal 55
Time Scaling 57
Robustness 58
SUMMARY 59
FURTHER READING 59

Chapter 4. Applications and Editing of Stored Voice 60

TAXONOMY OF VOICE OUTPUT APPLICATIONS 61
Playback-Only Applications 61
Interactive Record and Playback Applications 62
Dictation 63
Voice as a Document Type 64
VOICE IN INTERACTIVE DOCUMENTS 65

Ex.1018 / Page 8 of 343
AMAZON.COM, INC.

VOICE EDITING 69
Temporal Granularity 69
Manipulation of Audio Data 70
EXAMPLES OF VOICE EDITORS 74
Intelligent Ear, M.I.T. 74
Tioga Voice, Xerox PARC 75
PX Editor, Bell Northern Research 76
Sedit, Olivetti Research Center, and M.I.T. Media Laboratory 78
Pitchtool, M.I.T. Media Laboratory 79
SUMMARY 80

Chapter 5. Speech Synthesis 82

SYNTHESIZING SPEECH FROM TEXT 84
FROM TEXT TO PHONEMES 85
Additional Factors for Pronunciation 87
FROM PHONEMES TO SOUND 91
Parametric Synthesis 91
Concatenative Synthesis 93
QUALITY OF SYNTHETIC SPEECH 94
Measuring Intelligibility 95
Listener Satisfaction 96
Performance Factors 96
APPLICATIONS OF SYNTHETIC SPEECH 97
SUMMARY 99
FURTHER READING 99

Chapter 6. Interactive Voice Response 100

LIMITATIONS OF SPEECH OUTPUT 101
Speed 101
Temporal Nature 102
Serial Nature 102
Bulkiness 102
Privacy 103
ADVANTAGES OF VOICE 104

Ex.1018 / Page 9 of 343
AMAZON.COM, INC.

DESIGN CONSIDERATIONS 105
Application Appropriateness 106
Data Appropriateness 107
Responsiveness 108
Speech Rate 108
Interruption 109
Repetition 109
Exception Pronunciation 110
Multiple Voices 111
USER INPUT WITH TOUCHTONES 112
Menus 112
Data Entry 113
CASE STUDIES 117
Direction Assistance 117
Back Seat Driver 121
Voiced Mail 124
SUMMARY 130

Chapter 7. Speech Recognition 132

BASIC RECOGNIZER COMPONENTS 132
SIMPLE RECOGNIZER 133
Representation 134
Templates 134
Pattern Matching 135
CLASSES OF RECOGNIZERS 137
Who Can Use the Recognizer? 137
Speaking Style: Connected or Isolated Words? 139
Vocabulary Size 140
ADVANCED RECOGNITION TECHNIQUES 141
Dynamic Time Warping 142
Hidden Markov Models 144
Vector Quantization 147
Employing Constraints 149

Ex.1018 / Page 10 of 343
AMAZON.COM, INC.

ADVANCED RECOGNITION SYSTEMS 151
IBM's Tangora 151
CMU's Sphinx 151
MIT's SUMMIT 152
SUMMARY 152
FURTHER READING 153

Chapter 8. Using Speech Recognition 154

USES OF VOICE INPUT 154
Sole Input Channel 154
Auxiliary Input Channel 156
Keyboard Replacement 157
SPEECH RECOGNITION ERRORS 160
Classes of Recognition Errors 160
Factors Influencing the Error Rate 161
INTERACTION TECHNIQUES 163
Minimizing Errors 164
Confirmation Strategies 165
Error Correction 167
CASE STUDIES 169
Xspeak: Window Management by Voice 170
Put That There 175
SUMMARY 178

Chapter 9. Higher Levels of Linguistic Knowledge 179

SYNTAX 180
Syntactic Structure and Grammars 180
Parsers 185
SEMANTICS 186
PRAGMATICS 189
Knowledge Representation 190
Speech Acts 192
Conversational Implicature and Speech Acts 193

Ex.1018 / Page 11 of 343
AMAZON.COM, INC.

DISCOURSE 194
Regulation of Conversation 195
Discourse Focus 197
CASE STUDIES 199
Grunt 199
Conversational Desktop 204
SUMMARY 208
FURTHER READING 209

Chapter 10. Basics of Telephones 210

FUNCTIONAL OVERVIEW 211
ANALOG TELEPHONES 212
Signaling 213
Transmission 218
DIGITAL TELEPHONES 221
Signaling 222
Transmission 224
PBXS 226
SUMMARY 228
FURTHER READING 229

Chapter 11. Telephones and Computers 230

MOTIVATION 231
Access to Multiple Communication Channels 231
Improved User Interfaces 232
Enhanced Functionality 233
Voice and Computer Access 234
PROJECTS IN INTEGRATED TELEPHONY 234
Etherphone 234
MICE 237
BerBell 239
Personal eXchange 239
Phonetool 242

ARCHITECTURES 244
Distributed Architectures 244
Centralized Architectures 246
Comparison of Architectures 249
CASE STUDIES 251
Phone Slave 251
Xphone and Xrolo 256
Flexible Call Routing 260
SUMMARY 267

Chapter 12. Desktop Audio 268

EFFECTIVE DEPLOYMENT OF DESKTOP AUDIO 269
GRAPHICAL USER INTERFACES 271
AUDIO SERVER ARCHITECTURES 273
UBIQUITOUS AUDIO 278
CASE STUDIES 281
Evolution of a Visual Interface 282
Conversational Desktop 285
Phoneshell 287
Visual User Interfaces to Desktop Audio 292
SUMMARY 295

Chapter 13. Toward More Robust Communication 297

ROBUST COMMUNICATION 298
SPEECH RECOGNITION AND ROBUST PARSING 299
PROSODY 301
WHAT NEXT? 303

Bibliography 305
Index 315

Ex.1018 / Page 13 of 343
AMAZON.COM, INC.

Speaking of Talk

As we enter the next millennium, speech unquestionably will become the primary means of communication between human beings and machines. By primary I mean the one most often used. The reasons are simple and go beyond our intuitive desire to use native spoken language, which comes naturally to us and accounts for so much of human-to-human discourse.

Namely, the value of speech is not just the use of "natural language," as such, but includes other valuable properties. Speech works in the dark. It allows you to communicate with small objects. It lets you transmit a message beyond arm's reach and around corners. It is a subcarrier of meaning through prosody, tone and, of course, volume.[1]

How does speech play into today's trends in computing?

Miniaturization is surely one such trend. I expect to have more computing on my wrist in ten years than I have on my desk today. Another is ubiquity. I expect computing to be woven into the entire fabric of my life.[2] Another is concurrence. More and more, I see the parallel use of our auditory channels, when our hands and eyes are occupied.[3] Finally, I anticipate the increasing use of computers "in passing." By that I mean the ability to voice a remark, pose a question, or give a

[1]Any parent knows that it is not what you say, but how you say it.

[2]In the literal sense, as well, I expect apparel to be part of my private area network: PAN.

[3]Schmandt illustrates this in Chapter 6 with his example of the Back Seat Driver.

Ex.1018 / Page 15 of 343
AMAZON.COM, INC.

salutation without bringing all of one's other faculties to a grinding halt, as we do now when we sit down to a session with our computer. Computer usage today is solely in the fovea of our attention, while tomorrow we will enjoy it equally at the periphery of thoughts, actions, and consciousness, more like air than headlines.

Consider these trends and the features intrinsic to the audio domain, and there is little question of where we are headed. Talk will be the most common means of interaction with all machines, large and small.[4]

This book is not only timely but different. Chris Schmandt brings a unique and important perspective to speech. Let me illustrate, with a story, the difference I see between his work and the work of others. The story is old and has gained some color with time, but illustrates an important point about breaking engineering deadlocks.

In 1978, the so-called Architecture Machine Group received the first working model of NEC's connected speech recognizer—the top of the line for its day, designed by Dr. Yasuo Kato, now a member of the Board of Directors of that same corporation. It was a continuous speech recognition system, speaker-dependent, and able to handle fifty utterances.[5]

In the celebrated program called Put-That-There (discussed in Chapter 8) this machine worked with unmatched elegance and became the objective and site for technical tourists from all over the world. But it had a problem; one that turned out not to be unique to that specific device but, in fact, quite general.

If a user of the system was tense or nervous, his or her voice would be affected. Although this condition was not necessarily audible to another human being, it was distinctly noticeable to the machine. This seemingly harmless feature turned into a disastrous bug, since the enormous importance of presenting one's results to sponsors often creates tension.

The Advanced Research Projects Agency (ARPA) makes periodic visits to MIT, frequently with very high-ranking military personnel. Of course, ARPA was very interested in this elegant prototype of the future of command and control systems.[6] The importance of these visits was so great that graduate students, sometimes Chris Schmandt himself, became sufficiently nervous that a perfectly working system (ten minutes before) functioned improperly in the presence of these august, easily disappointed, and frequently disbelieving visitors. There is something particularly frustrating and stupid looking about a speech recognition system when it does not work, which only exacerbates the problem and makes the person demonstrating the system even more anxious.

[4]I do not limit my remarks to computers, as we know them, but consider all machines: home appliances, entertainment equipment, telecommunication apparati, transportation machinery, engines for work, play, and learning—as well as some not yet imagined.

[5]The difference between a word and an utterance is crucial, as an utterance can be a part of a word or a concatenation of them (up to five seconds in this particular case).

[6]Command and control is deeply rooted in voice; my own limited experience in Army exercises witnessed a great deal of yelling.

Ex.1018 / Page 16 of 343
AMAZON.COM, INC.

The short-term fix was to train students and staff who demonstrated the system to be cool, calm, and collected and to ignore the stature of their guests. Guests were encouraged to wear blue jeans versus uniforms.

The longer-term solution resides in a different view of speech, as communication and interaction, not just as a string of words in the acoustic domain. This perspective results in a variety of ideas, of which one is to find the pauses in the speaker's speech[7] and judiciously issue the utterance "ah ha," taken from a suite of "ah ha's" ranging from the mousy to the elephantine. Namely, as one speaks to a system it should listen and, now and again, say "ah ha"—"ah ha"—"ah ha." The user will relax and the system's performance will skyrocket.

Discussions of ideas such as this have been highly criticized as yet another example of how members of the Media Lab are mere charlatans. How dare they give a user the false impression that the machine has understood when, in fact, it may not have?

The criticism is a perfect example of why computer science is so sensory deprived and getting duller by the day. Critics miss the point totally. Listen to yourself on the telephone. As someone is talking to you, you will naturally issue an "ah ha" every fifteen to thirty seconds. In fact, if you don't, after a while your telephone partner will be forced to ask, "Are you still there?" The point is that the "ah ha" *has no lexical value*. It is purely a communications protocol, no different from the electronic hand-shaking with which network architects are so familiar.

How did this escape researchers in the speech recognition community, itself a mature pursuit for over twenty-five years? The same question can be rephrased: Why was speech production and speech recognition so utterly decoupled?[8]

My answer is not popular with most members of the speech community, but it is at the root of why this book is so special. That is, the traditional speech recognition community was fundamentally disinterested in "communication." Its interest was in "transcription"; namely, transcoding from ASCII into utterances or spoken language into ASCII. It was not interested in the tightly coupled interaction of human dialogue, typical of face-to-face communication. That kind of speech, Schmandt's kind of speech, is filled with interruption, retort, and paralinguals (the "ah ha's"), wherein the verbal back-channel is no less important than speech output. This concept is what *Voice Communication with Computers* brings to the table.

Voice Communication with Computers is not just a textbook or tutorial, it is a perspective. Its point of view underscores the need to interact and communicate with computers in much the same way we communicate with human beings. Only when we are able to talk to machines the way we talk to people will the concept of "ease of use" be seriously addressed.

[7]We all pause, even those of us who speak like machine guns must come up for air.

[8]Literally, in almost all labs world-wide, researchers in speech production and speech recognition were in different groups, different labs, or different divisions.

Ex.1018 / Page 17 of 343
AMAZON.COM, INC.

Today "ease of use" means simple, direct manipulation, with little or no delegation. The future of computing is discourse and delegation and, for that, speech is mandatory—not a process of transcription but conversation. This is the area pioneered by Schmandt and reflected in the following pages.

Nicholas Negroponte
MIT Media Lab

Ex.1018 / Page 18 of 343
AMAZON.COM, INC.

Preface

This book began as graduate class notes to supplement technical readings from a number of disciplines. Over the years as I have worked on various projects, I kept running against the question, "What can I read to better understand this?" Although materials are available, texts have been too focused and detailed and papers are spread across too many conference proceedings for either students or specialists in other areas to make good use of them. This book is meant to open the more specialized disciplines to a wider audience while conveying a vision of speech communication integrated with computing.

The primary message of this book is an approach to incorporating speech into computing environments; it is unique in its focus on the *user* of speech systems and interaction techniques to enhance the utility of voice computing. But a complete evaluation of speech at the interface is grounded in the understanding of speech technologies, the computational means of expressing linguistic and discourse knowledge, and the software architectures that support voice in multimedia computer environments.

This interdisciplinary approach is the most valuable asset of this book but simultaneously a limitation. I attempt to open technical doors for readers who span many fields and professions. For example, I discuss speech coding without reference to the standard notations of discrete time digital signal processing. This is a little bit like trying to teach high school physics to students who have not yet learned calculus: terribly inelegant in retrospect *if* one pursues studies in physics. But this is not a speech coding text; this book examines how and when to *use* speech coding, speech recognition, text-to-speech synthesis, computer-telephone interfaces, and natural language processing. Anyone trying to under-

Ex.1018 / Page 19 of 343
AMAZON.COM, INC.

stand how to use voice in emerging computing environments should benefit from this information.

Technology has changed rapidly in the five years this book has been in progress. While it has been reassuring to see many of my expectations fulfilled—and technological developments only increase the potential for pervasive deployment of speech-capable applications—these changes have necessitated continual updating in this text. This pace of change almost guarantees that some portion of this book will be out of date by the time it appears in print. Five years ago speech technology was awkward, expensive, and difficult to integrate into work environments. Speech systems research almost invariably required interfacing to additional peripheral devices including complete components such as text-to-speech synthesizers or internal add-in hardware such as digital signal processors. No computer was equipped with a microphone back then, and while computer speakers usually were connected to tone-generating hardware to emit various beep sounds, these systems could not play a digital audio file. At that time, large-scale consumer access to digitized audio via the compact disc was only just beginning to make serious inroads on analog audio tape and vinyl LP music distribution.

Much has changed while I wrote and rewrote this book. Most computers are now audio capable, equipped with microphones and the ability to play back digital audio at qualities ranging from that of the telephone to that of the CD. The capacity of disk drives is now measured in Gigabytes, not Megabytes, minimizing the difficulty of storing audio data, which requires much more disk space than text. Some computers have been shipped with digital signal microprocessors included on the mother board. Others ship with digital-telephone network interfaces (ISDN) as standard equipment. The explosion of cellular telephony and palmtop computers has created a new community of potential computer audio users, who need the access and the means to communicate with computers as well as with other people while on the move.

Perhaps the most significant development, however, is the more than ten-fold increase in the speed of general-purpose processors found in all classes of computers. This speed now enables the majority of computers to perform basic real-time voice processing and simple speech recognition. The more powerful desktop workstations are capable of the highest quality text-to-speech synthesis, large vocabulary speech recognition, and sophisticated voice processing. Speech technology is rapidly becoming a commodity that can be distributed as shrink-wrapped software or included with a computer operating system.

During these five years, applications have also followed the enabling technology although with a frustrating lag. Basic speech recognition to control computer window systems is available in several products. A number of isolated applications allow voice annotations to spreadsheets and other text-oriented data. Several competing formats for voice attachments to electronic mail have also appeared, and now progress is being made to converge on a single standard format. Recently a new venture initiated a digital audio "radio" program distribution over the Internet. Speech recognition is now being used in the telephone network, to help complete collect call attempts, for example.

Ex.1018 / Page 20 of 343
AMAZON.COM, INC.

This is just the beginning of ubiquitous voice technology. Voice is so useful and such an important part of our lives that, by necessity, it will become pervasive in the ways computers are used. The trends of the last five years will continue only to accelerate over the next five. While we must accept that we are still a long, long way from the science fiction theme of the ever-present listening computer (with a pleasant voice as well!), we will find ourselves holding an increasing number of conversations with machines. Some of the readers of this book will be among those researchers or developers making this a reality.

Ex.1018 / Page 21 of 343
AMAZON.COM, INC.

Acknowledgments

This book originated with sketchy class notes coauthored with Jim Davis, and Pascal Chesnais triggered our note-writing by encouraging us to give students almost anything. Jim wisely decided to complete his dissertation instead of joining me to write this book and graduated years ago.

I have had help from many people both in editing my technical writing as well as reviewing portions of this manuscript for accuracy. I have benefitted tremendously from their help; the errors that remain are my own. This list includes Barry Arons, Janet Cahn, Jim Davis, Kate Gordon, Harry Hersh, Lisa Stifelman, Ben Stoltz, Tony Vitale, and Nicole Yankelovich.

I have also benefitted tremendously from both editorial help as well as immeasurable assistance with the production of this book from Gayle Sherman and Elaine McCarthy. I think I would have given up months from completion without Gayle's help. Without the patience of Dianne Littwin at Van Nostrand Reinhold, I'm sure I would have backed out of this project years ago. And most important has been the support of my family, Ava and Kaya, who let me add the time to write this book to an already overly crowded work schedule.

This book includes descriptions of many research projects we have implemented at M.I.T, both at the Media Laboratory as well as one of its predecessors, the Architecture Machine Group. I would never have been in the position to create the group that performed this work without the unwavering support of my laboratory director, Nicholas Negroponte, for the last thirteen years. And I probably would have been unable to deal with the politics of a new laboratory without the comradeship of Walter Bender, even though we have worked on completely independent projects since jointly debugging network code a decade ago.

Ex.1018 / Page 23 of 343
AMAZON.COM, INC.

Finally and most importantly, my own ideas as well as these projects have evolved over years of interacting with students. They have written much of the software, helped me debug my class notes, and suffered through my attempts to put these thoughts into a coherent whole in my courses. Some have been friends, some have been peers; yet we have had plenty of disagreements. But our graduates are more valuable than anything else we do, and I have enjoyed working with them immensely. It would be impossible to rank-order their contributions, so I list them alphabetically: Mark Ackerman, Barry Arons, Derek Atkins, Lorne Berman, Jim Davis, Lorin Jurow, Debby Hindus, Angie Hinrichs, Chris Horner, Eric Hulteen, Andrew Kass, Kevin Landel, Eric Ly, Sanjay Manandhar, Mike McKenna, Atty Mullins, Sheldon Pacotti, Charles Simmons, Jordan Slott, Lisa Stifelman, Tom Trobaugh, Mark Vershel, Todd Wayne, Chi Wong, and Jim Zamiska.

During the course of writing this book I discovered the deserts of the American Southwest. Over the years I find myself repeatedly drawn back to their open spaces and infinite vistas. Although I have edited a significant portion of this book in their midst I have missed them badly while sitting at desks trying to put this project to rest. Although the two are somehow inextricably intermingled in my consciousness, I look forward to getting back to the purity of the desert without carrying this project along.

Ex.1018 / Page 24 of 343
AMAZON.COM, INC.

Introduction

For most of us, speech has been an integral part of our daily lives since we were small children. Speech *is* communication; it is highly expressive and conveys subtle intentions clearly. Our conversations employ a range of interactive techniques to facilitate mutual understanding and ensure that we are understood.

But despite the effectiveness of speech communication, few of us use speech in our daily computing environments. In most workplaces voice is relegated to specialized industrial applications or aids to the disabled; voice is not a part of the computer interfaces based on displays, keyboards, and mice. Although current workstations have become capable of supporting much more sophisticated voice processing, the most successful speech application to date, voice mail, is tied most closely to the telephone.

As speech technologies and natural language understanding mature in the coming decades, many more potential applications will become reality. But much more than raw technology is required to bridge the gap between human conversation and computer interfaces; we must understand the assets and liabilities of voice communication if we are to gauge under which circumstances it will prove to be valuable to end users.

Conversational systems must speak and listen, but they also must understand, pose queries, take turns, and remember the topic of conversation. Understanding how people converse lets us develop better models for interaction with computers by voice. But speech is a very demanding medium to employ effectively, and unless user interaction techniques are chosen with great care, voice applications tend to be slow and awkward to use.

Ex.1018 / Page 25 of 343
AMAZON.COM, INC.

This book is about using speech in a variety of computing environments based on appreciating its role in human communication. Speech can be used as a method of *interacting* with a computer to place requests or receive warnings and notices. Voice can also be used as the underlying *data* itself, such as notes stored in a calendar, voice annotations of a text document, or telephone messages. Desktop workstations can already support both these speech functions. Speech excels as a method of interacting with the desktop computer over the telephone and has strong potential as the primary channel to access a computer small enough to fit in one's shirt pocket. The full utility of speech will be realized only when it is integrated across *all* these situations; when users find it effective to talk to their computers over the telephone, for example, they will suddenly have more utility for voice as data while in the office.

CONTENTS OF THIS BOOK

This book serves different needs for different readers. The author believes that a firm grounding in the theory of operation of speech technologies forms an important basis for appreciating the difficulties of building applications and interfaces to employ them. This understanding is necessary if we wish to be capable of making any predictions or even guesses of where this field will lead us over the next decade. Paired with descriptions of voice technologies are chapters devoted to applications and user interaction techniques for each, including case studies to illustrate potential applications in more detail. But many chapters stand more or less on their own, and individual readers may pick and choose among them. Readers interested primarily in user interface design issues will gain most benefit from Chapters 4, 6, 8, 9, and 12. Those most concerned about system architectures and support for voice in multimedia computing environments should focus on Chapters 3, 5, 7, and 12. A telecommunications perspective is the emphasis of Chapters 10, 11, and 6.

A conversation requires the ability to speak and to listen, and, if the parties are not in close proximity, some means of transporting their voices across a distance. Chapter 1 discusses the communicative role of speech and introduces some representations of speech and an analytic approach that frames the content of this book. Chapter 2 discusses the physiology of human speech and how we perceive it through our ears; although later chapters refer back to this information, it is not essential for understanding the remainder of the book.

Voice interface technologies are required for computers to participate in conversations. These technologies include digital recording, speech synthesis, and speech recognition; these are the topics of Chapters 3, 5, and 7. Knowledge of the operations of the speech technologies better prepares the reader to appreciate their limitations and understand the impact of improvements in the technologies in the near and distant future.

Although speech is intuitive and seemingly effortless for most of us, it is actually quite difficult to employ as a computer interface. This difficulty is partially due to limitations of current technology but also a result of characteristics inher-

Ex.1018 / Page 26 of 343
AMAZON.COM, INC.

ent in the speech medium itself. The heart of this book is both criteria for evaluating the suitability of voice to a range of applications and interaction techniques to make its use effective in the user interface. Although these topics are treated throughout this book, they receive particular emphasis in Chapters 4, 6, 8 and 12. These design guidelines are accentuated by case studies scattered throughout the book but especially in these chapters.

These middle chapters are presented in pairs. Each pair contains a chapter describing underlying technology matched with a chapter discussing how to apply the technology. Chapter 3 describes various speech coding methods in a descriptive form and differentiates coding schemes based on data rate, intelligibility, and flexibility. Chapter 4 then focuses on simple applications of stored voice in computer documents and the internal structure of audio editors used to produce those documents. Chapter 5 introduces text-to-speech algorithms. Chapter 6 then draws on both speech coding as well as speech synthesis to discuss *interactive* applications using speech output over the telephone.

Chapter 7 introduces an assortment of speech recognition techniques. After this, Chapter 8 returns to interactive systems, this time emphasizing voice input instead of touch tones. The vast majority of work to date on systems that speak and listen has involved short utterances and brief transactions. But both sentences and conversations exhibit a variety of structures that must be mastered if computers are to become *fluent.* Syntax and semantics constrain sentences in ways that facilitate interpretation; pragmatics relates a person's utterances to intentions and real-world objects; and discourse knowledge indicates how to respond and carry on the thread of a conversation across multiple exchanges. These aspects of speech communication, which are the focus of Chapters 9 and 13, must be incorporated into any system that can engage successfully in a conversation that in any way approaches the way we speak to each other.

Although a discussion of the workings of the telephone network may at first seem tangential to a book about voice in computing, the telephone plays a key role in any discussion of speech and computers. The ubiquity of the telephone assures it a central role in our voice communication tasks. Every aspect of telephone technology is rapidly changing from the underlying network to the devices we hold in our hands, and this is creating many opportunities for computers to get involved in our day-to-day communication tasks. Chapter 10 describes the telephone technologies, while Chapter 11 discusses the integration of telephone functionality into computer workstations. Much of Chapter 6 is about building telephone-based voice applications that can provide a means of accessing personal databases while not in the office.

When we work at our desks, we may employ a variety of speech processing technologies in isolation, but the full richness of voice at the desktop comes with the combination of multiple voice applications. Voice applications on the workstation also raise issues of interaction between both audio and window systems and operating system and run-time support for voice. This is the topic of Chapter 12. Speakers and microphones at every desk may allow us to capture many of the spontaneous conversations we hold every day, which are such an essential

Ex.1018 / Page 27 of 343
AMAZON.COM, INC.

aspect of our work lives. Desktop voice processing also enables remote telephone access to many of the personal information management utilities that we use in our offices.

ASSUMPTIONS

This book covers material derived from a number of specialized disciplines in a way that is accessible to a general audience. It is divided equally between background knowledge of speech technologies and practical application and interaction techniques. This broad view of voice communication taken in this book is by definition interdisciplinary. Speech communication is so vital and so rich that a number of specialized areas of research have risen around it, including speech science, digital signal processing and linguistics, aspects of artificial intelligence (computational linguistics), cognitive psychology, and human factors. This book touches on all these areas but makes no pretense of covering any of them in depth. This book attempts to open doors by revealing why each of these research areas is relevant to the design of conversational computer systems; the reader with further interest in any of these fields is encouraged to pursue the key overview references mentioned in each chapter.

Significant knowledge of higher mathematics as well as digital signal processing is assumed by many speech texts. These disciplines provide an important level of abstraction and on a practical level are tools required for any serious development of speech technology itself. But to be accessible to a wider audience, this book makes little use of mathematics beyond notation from basic algebra. This book provides an intuitive, rather than rigorous, treatment of speech signal processing to aid the reader in evaluation and selection of technologies and to appreciate their operation and design tradeoffs.

There is a wide gap between the goal of emulating conversational human behavior and what is commercially viable with today's speech technology. Despite the large amount of basic speech research around the world, there is little innovative work on how speech devices may be used in advanced systems, but it is difficult to discuss applications without examples. To this end, the author has taken the liberty to provide more detail with a series of voice projects from the Speech Research Group of M.I.T.'s Media Laboratory (including work from one of its predecessors, the Architecture Machine Group). Presented as case studies, these projects are intended both to illustrate applications of the ideas presented in each chapter and to present pertinent design issues. It is hoped that taken collectively these projects will offer a vision of the many ways in which computers can take part in communication.

Ex.1018 / Page 28 of 343
AMAZON.COM, INC.

1

Speech as Communication

Speech can be viewed in many ways. Although chapters of this book focus on specific aspects of speech and the computer technologies that utilize speech, the reader should begin with a broad perspective on the role of speech in our daily lives. It is essential to appreciate the range of capabilities that conversational systems must possess before attempting to build them. This chapter lays the groundwork for the entire book by presenting several perspectives on speech communication.

The first section of this chapter emphasizes the *interactive* and *expressive* role of voice communication. Except in formal circumstances such as lectures and dramatic performances, speech occurs in the context of a *conversation,* wherein participants take turns speaking, interrupt each other, nod in agreement, or try to change the topic. Computer systems that talk or listen may ultimately be judged by their ability to converse in like manner simply because conversation permeates human experience. The second section discusses the various components or *layers* of a conversation. Although the distinctions between these layers are somewhat contrived, they provide a means of analyzing the communication process; research disciplines have evolved for the study of each of these components. Finally, the last section introduces the *representations* of speech and conversation, corresponding in part to the layers identified in the second section. These representations provide abstractions that a computer program may employ to engage in a conversation with a human.

Ex.1018 / Page 29 of 343
AMAZON.COM, INC.

SPEECH AS CONVERSATION

Conversation is a process involving multiple participants, shared knowledge, and a protocol for taking turns and providing mutual feedback. Voice is our primary channel of interaction in conversation, and speech evolved in humans in response to the need among its members to communicate. It is hard to imagine many uses of speech that do not involve some interchange between multiple participants in a conversation; if we are discovered talking to ourselves, we usually feel embarrassed.

For people of normal physical and mental ability, speech is both rich in expressiveness and easy to use. We learn it without much apparent effort as children and employ it spontaneously on a daily basis.[1] People employ many layers of knowledge and sophisticated protocols while having a conversation; until we attempt to analyze dialogues, we are unaware of the complexity of this interplay between parties.

Although much is known about language, study of interactive speech communication has begun only recently. Considerable research has been done on natural language processing systems, but much of this is based on keyboard input. It is important to note the contrast between written and spoken language and between read or rehearsed speech and spontaneous utterances. Spoken language is less formal than written language, and errors in construction of spoken sentences are less objectionable. Spontaneous speech shows much evidence of the real-time processes associated with its production, including false starts, nonspeech noises such as mouth clicks and breath sounds, and pauses either silent or filled (". . . um . . .") [Zue *et al.* 1989b]. In addition, speech naturally conveys intonational and emotional information that fiction writers and playwrights must struggle to impart to written language.

Speech is rich in interactive techniques to guarantee that the listener understands what is being expressed, including facial expressions, physical and vocal gestures, "uh-huhs," and the like. At certain points in a conversation, it is appropriate for the listener to begin speaking; these points are often indicated by longer pauses and lengthened final syllables or marked decreases in pitch at the end of a sentence. Each round of speech by one person is called a **turn; interruption** occurs when a participant speaks before a break point offered by the talker. Instead of taking a turn, the listener may quickly indicate agreement with a word or two, a nonverbal sound ("uh-huh"), or a facial gesture. Such responses, called **back channels,** speed the exchange and result in more effective conversations [Kraut *et al.* 1982].[2]

Because of these interactive characteristics, speech is used for immediate communication needs, while writing often implies a distance, either in time or space,

[1]For a person with normal speech and hearing to spend a day without speaking is quite a novel experience.

[2]We will return to these topics in Chapter 9.

Ex.1018 / Page 30 of 343
AMAZON.COM, INC.

between the author and reader. Speech is used in transitory interactions or situations in which the process of the interaction may be as important as its result. For example, the agenda for a meeting is likely to be written, and a written summary or minutes may be issued "for the record," but the actual decisions are made during a conversation. Chapanis and his colleagues arranged a series of experiments to compare the effectiveness of several communication media, i.e., voice, video, handwriting, and typewriting, either alone or in combination, for problem-solving tasks [Ochsman and Chapanis 1974]. Their findings indicated an overwhelming contribution of voice for such interactions. Any experimental condition that included voice was superior to any excluding voice; the inclusion of other media with voice resulted in only a small additional effectiveness. Although these experiments were simplistic in their use of student subjects and invented tasks and more recent work by others [Minneman and Bly 1991] clarifies a role for video interaction, the dominance of voice seems unassailable.

But conversation is more than mere interaction; communication often serves a purpose of changing or influencing the parties speaking to each other. I tell you something I have learned with the intention that you share my knowledge and hence enhance your view of the world. Or I wish to obtain some information from you so I ask you a question, hoping to elicit a reply. Or perhaps I seek to convince you to perform some activity for me; this may be satisfied either by your physical performance of the requested action or by your spoken promise to perform the act at a later time. "Speech Act" theories (to be discussed in more detail in Chapter 9) attempt to explain language as action, e.g., to request, command, query, and promise, as well as to inform.

The intention behind an utterance may not be explicit. For example, "Can you pass the salt?" is not a query about one's ability; it is a request. Many actual conversations resist such purposeful classifications. Some utterances ("go ahead," "uh-huh," "just a moment") exist only to guide the flow of the conversation or comment on the state of the discourse, rather than to convey information. Directly purposeful requests are often phrased in a manner allowing flexibility of interpretation and response. This looseness is important to the process of people defining and maintaining their work roles with respect to each other and establishing socially comfortable relationships in a hierarchical organization. The richness of speech allows a wide range of "acceptance" and "agreement" from wholehearted to skeptical to incredulous.

Speech also serves a strong social function among individuals and is often used just to pass the time, tell jokes, or talk about the weather. Indeed, extended periods of silence among a group may be associated with interpersonal awkwardness or discomfort. Sometimes the actual occurrence of the conversation serves a more significant purpose than any of the topics under discussion. Speech may be used to call attention to oneself in a social setting or as an exclamation of surprise or dismay in which an utterance has little meaning with respect to any preceding conversation. [Goffman 1981]

The expressiveness of speech and robustness of conversation strongly support the use of speech in computer systems, both for stored voice as a data type as well as speech as a medium of interaction. Unfortunately, current computers are

Ex.1018 / Page 31 of 343
AMAZON.COM, INC.

capable of uttering only short sentences of marginal intelligibility and occasionally recognizing single words. Engaging a computer in a conversation can be like an interaction in a foreign country. One studies the phrase book, utters a request, and in return receives either a blank stare (wrong pronunciation, try again) or a torrent of fluent speech in which one cannot perceive even the word boundaries.

However, limitations in technology only reinforce the need to take advantage of conversational techniques to ensure that the user is understood. Users will judge the performance of computer systems employing speech on the basis of their expectations about conversation developed from years of experience speaking with fellow humans. Users may expect computers to be either deaf and dumb, or once they realize the system can talk and listen, expect it to speak fluently like you and me. Since the capabilities of current speech technology lie between these extremes, building effective conversational computer systems can be very frustrating.

HIERARCHICAL STRUCTURE OF CONVERSATION

A more analytic approach to speech communication reveals a number of different ways of describing what actually occurs when we speak. The hierarchical structure of such analysis suggests goals to be attained at various stages in computer-based speech communication.

Conversation requires apparatus both for listening and speaking. Effective communication invokes mental processes employing the mouth and ears to convey a message thoroughly and reliably. There are many layers at which we can analyze the communication process, from the lower layers where speech is considered primarily acoustically to higher layers that express meaning and intention. Each layer involves increased knowledge and potential for intelligence and interactivity.

From the point of view of the speaker, we may look at speech from at least eight layers of processing as shown in Figure 1.1.

Layers of Speech Processing

discourse The regulation of conversation for pragmatic ends. This includes taking turns talking, the history of referents in a conversation so pronouns can refer to words spoken earlier, and the process of introducing new topics.

pragmatics The intent or motivation for an utterance. This is the underlying reason the utterance was spoken.

semantics The meaning of the words individually and their meaning as combined in a particular sentence.

syntax The rules governing the combination of words in a sentence, their parts of speech, and their forms, such as case and number.

Ex.1018 / Page 32 of 343
AMAZON.COM, INC.

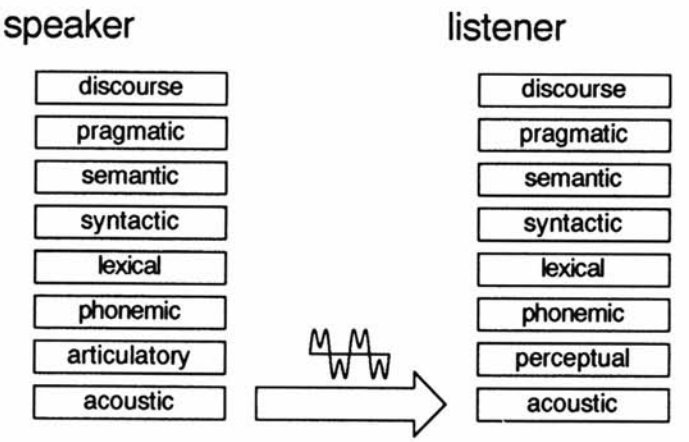


Figure 1.1. A layered view of speech communication.

lexical The set of words in a language, the rules for forming new words from affixes (prefixes and suffixes), and the stress ("accent") of syllables within the words.

phonetics The series of sounds that uniquely convey the series of words in the sentence.

articulation The motions or configurations of the vocal tract that produce the sounds, e.g., the tongue touching the lips or the vocal cords vibrating.

acoustics The realization of the string of phonemes in the sentence as vibrations of air molecules to produce pressure waves, i.e., sound.

Consider two hikers walking through the forest when one hiker's shoelace becomes untied. The other hiker sees this and says, "Hey, you're going to trip on your shoelace." The listener then ties the shoelace. We can consider this utterance at each layer of description.

Discourse analysis reveals that "Hey" serves to call attention to the urgency of the message and probably indicates the introduction of a new topic of conversation. It is probably spoken in a raised tone and an observer would reasonably expect the listener to acknowledge this utterance, either with a vocal response or by tying the shoe. Experience with discourse indicates that this is an appropriate interruption or initiation of a conversation at least under some circumstances. Discourse structure may help the listener understand that subsequent utterances refer to the shoelace instead of the difficulty of the terrain on which the conversants are traveling.

In terms of **pragmatics,** the speaker's intent is to warn the listener against tripping; presumably the speaker does not wish the listener to fall. But this utterance might also have been a ruse intended to get the listener to look down for the sake of playing a trick. We cannot differentiate these possibilities without know-

Ex.1018 / Page 33 of 343
AMAZON.COM, INC.

ing more about the context in which the sentence was spoken and the relationship between the conversants.

From a **semantics** standpoint, the sentence is about certain objects in the world: the listener, hiking, an article of clothing worn on the foot, and especially the string by which the boot is held on. The concern at the semantic layer is how the words refer to the world and what states of affairs they describe or predict. In this case, the meaning has to do with an animate entity ("you") performing some physical action ("tripping"), and the use of future tense indicates that the talker is making a prediction of something not currently taking place. Not all words refer to specific subjects; in the example, "Hey" serves to attract attention, but has no innate meaning.

Syntax is concerned with how the words fit together into the structure of the sentence. This includes the ordering of parts of speech (nouns, verbs, adjectives) and relations between words and the words that modify them. Syntax indicates that the correct word order is subject followed by verb, and syntax forces agreement of number, person, and case of the various words in the sentence. "You *is* going to . . ." is syntactically ill formed. Because the subject of the example is "you," the associated form of the verb "to be" is "are." The chosen verb form also indicates a future tense.

Lexical analysis tells us that "shoelace" comes from the root words "shoe" and "lace" and that the first syllable is stressed. Lexical analysis also identifies a set of definitions for each word taken in isolation. "Trip," for example, could be the act of falling or it could refer to a journey. Syntax reveals which definition is appropriate as each is associated with the word "trip" used as a different part of speech. In the example, "trip" is used as a verb and so refers to falling.

The **phonemic** layer is concerned with the string of phonemes of which the words are composed. Phonemes are the speech sounds that form the words of any language.[3] Phonemes include all the sounds associated with vowels and consonants. A grunt, growl, hiss, or gargling sound is not a phoneme in English, so it cannot be part of a word; such sounds are not referred to as speech. At the phoneme layer, while talking we are either continuously producing speech sounds or are silent. We are not silent at word boundaries; the phonemes all run together.

At the **articulatory** layer, the speaker makes a series of vocal gestures to produce the sounds that make up the phonemes. These sounds are created by a noise source at some location in the vocal tract, which is then modified by the configuration of the rest of the vocal tract. For example, to produce a "b" sound, the lips are first closed and air pressure from the lungs is built up behind them. A sudden release of air between the lips accompanied by vibration of the vocal cords produces the "b" sound. An "s" sound, by comparison, is produced by turbulence caused as a stream of air rushes through a constriction formed by the tongue and the roof of the mouth. The mouth can also be used to create nonspeech sounds, such as sighs and grunts.

[3]A more rigorous definition will be given in the next section.

Ex.1018 / Page 34 of 343
AMAZON.COM, INC.

Finally, the **acoustics** of the utterance is its nature as sound. Sound is transmitted as variations in air pressure over time; sound can be converted to an electrical signal by a microphone and represented as an electrical waveform. We can also analyze sound by converting the waveform to a spectrogram, which displays the various frequency components present in the sound. At the acoustic layer, speech is just another sound like the wind in the trees or a jet plane flying overhead.

From the perspective of the listener, the articulatory layer is replaced by a **perceptual** layer, which comprises the processes whereby sound (variations in air pressure over time) is converted to neural signals in the ear and ultimately interpreted as speech sounds in the brain. It is important to keep in mind that the hearer can directly sense only the acoustic layer of speech. If we send an electric signal representing the speech waveform over a telephone line and convert this signal to sound at the other end, the listening party can understand the speech. Therefore, the acoustic layer alone must contain all the information necessary to understand the speaker's intent, but it can be represented at the various layers as part of the process of understanding.

This layered approach is actually more descriptive than analytic in terms of human cognitive processes. The distinctions between the layers are fuzzy, and there is little evidence that humans actually organize discourse production into such layers. Intonation is interpreted in parallel at all these layers and thus illustrates the lack of sharp boundaries or sequential processing among them. At the pragmatic layer, intonation differentiates the simple question from exaggerated disbelief; the same words spoken with different intonation can have totally different meaning. At the syntactic layer, intonation is a cue to phrase boundaries. Intonation can differentiate the noun and verb forms of some words (e.g., conduct, convict) at the syntactic layer by conveying lexical stress. Intonation is not phonemic in English, but in some other languages a change in pitch does indicate a different word for the otherwise identical articulation. And intonation is articulated and realized acoustically in part as the fundamental frequency at which the vocal cords vibrate.

Dissecting the communication process into layers offers several benefits, both in terms of understanding as well as for practical implementations. Understanding this layering helps us appreciate the complexity and richness of speech. Research disciplines have evolved around each layer. A layered approach to representing conversation is essential for modular software development; a clean architecture isolates each module from the specialized knowledge of the others with information passed over well-defined interfaces. Because each layer consists of a different perspective on speech communication, each is likely to employ its own representation of speech for analysis and generation.

As a cautionary note, it needs to be recognized from the start that there is little evidence that humans actually function by invoking each of these layers during conversation. The model is descriptive without attempting to explain or identify components of our cognitive processes. The model is incomplete in that there are some aspects of speech communication that do not fit it, but it can serve as a framework for much of our discussion of conversational computer systems.

Ex.1018 / Page 35 of 343
AMAZON.COM, INC.

REPRESENTATIONS OF SPEECH

We need a means of describing and manipulating speech at each of these layers. Representations for the lower layers, such as acoustic waveforms or phonemes, are simpler and more complete and also more closely correspond to directly observable phenomena. Higher-layer representations, such as semantic or discourse structure, are subject to a great deal more argument and interpretation and are usually abstractions convenient for a computer program or a linguistic comparison of several languages. Any single representation is capable of conveying particular aspects of speech; different representations are suitable for discussion of the different layers of the communication process.

The representation chosen for any layer should contain all the information required for analysis at that layer. Since higher layers possess a greater degree of abstraction than lower layers, higher-layer representations extract features from lower-layer representations and hence lose the ability to recreate the original information completely. For example, numerous cues at the acoustic layer may indicate that the talker is female, but if we represent the utterance as a string of phones or of words we have lost those cues. In terms of computer software, the representation is the data type by which speech is described. One must match the representation and the particular knowledge about speech that it conveys to the algorithms employing it at a particular layer of speech understanding.

Acoustic Representations

Sounds consist of variations in air pressure over time at frequencies that we can hear. Speech consists of a subset of the sounds generated by the human vocal tract. If we wish to analyze a sound or save it to hear again later, we need to capture the variations in air pressure. We can convert air pressure to electric voltage with a microphone and then convert the voltage to magnetic flux on an audiocassette tape using a recording head, for example.

We can plot the speech signal in any of these media (air pressure, voltage, or magnetic flux) over time as a **waveform** as illustrated in Figure 1.2. This representation exhibits positive and negative values over time because the speech radiating from our mouths causes air pressure to be temporarily greater or less than that of the ambient air.

A waveform describing sound pressure in air is continuous, while the waveforms employed by computers are **digital,** or **sampled,** and have discrete values for each sample; these concepts are described in detail in Chapter 3. Tape recorders store analog waveforms; a compact audio disc holds a digital waveform. A digitized waveform can be made to very closely represent the original sound, and it can be captured easily with inexpensive equipment. A digitized sound stored in computer memory allows for fast random access. Once digitized, the sound may be further processed or compressed using digital signal processing techniques. The analog audiotape supports only sequential access (it must be rewound or fast-forwarded to jump to a different part of the tape) and is prone to mechanical breakdown.

Ex.1018 / Page 36 of 343
AMAZON.COM, INC.

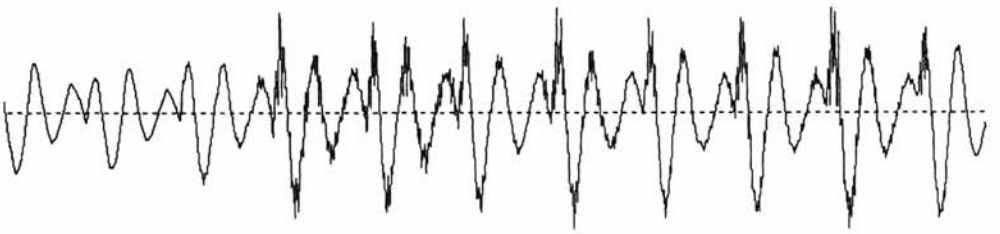


Figure 1.2. A waveform, showing 100 milliseconds of the word "me." The vertical axis depicts amplitude, and the horizontal axis represents time. The display depicts the transition from "m" to "e."

A waveform can effectively represent the original signal visually when plotted on a piece of paper or a computer screen. But to make observations about what a waveform sounds like, we must analyze it across a span of time not just at a single point. For example, in Figure 1.2 we can determine the amplitude ("volume") of the signal by looking at the differences between its highest and lowest points. We can also see that it is periodic: The signal repeats a pattern over and over. Since the horizontal axis represents time, we can determine the frequency of the signal by counting the number of periods in one second. A periodic sound with a higher frequency has a higher pitch than a periodic sound with a lower frequency.

One disadvantage of working directly with waveforms is that they require considerable storage space, making them bulky; Figure 1.2 shows only 100 milliseconds of speech. A variety of schemes for compressing speech to minimize storage are discussed in Chapter 3. A more crucial limitation is that a waveform simply shows the signal as a function of time. A waveform is in no way speech specific and can represent any acoustical phenomenon equally well. As a general-purpose representation, it contains all the acoustic information but does not explicitly describe its content in terms of properties of speech signals.

A speech-specific representation more succinctly conveys those features salient to speech and phonemes, such as syllable boundaries, fundamental frequency, and the higher-energy frequencies in the sound. A **spectrogram** is a transformation of the waveform into the frequency domain. As seen in Figure 1.3, the spectrogram reveals the distribution of various frequency components of the signal as a function of time indicating the energy at each frequency. The horizontal axis represents time, the vertical axis represents frequency, and the intensity or blackness at a point indicates the acoustic energy at that frequency and time.

A spectrogram still consists of a large amount of data but usually requires much less storage than the original waveform and reveals acoustic features specific to speech. Because of this, spectral analysis[4] is often employed to process

[4]To be precise, spectral or Fourier analysis uses mathematical techniques to derive the values of energy at particular frequencies. We can plot these as described above; this visual representation is the spectrogram.

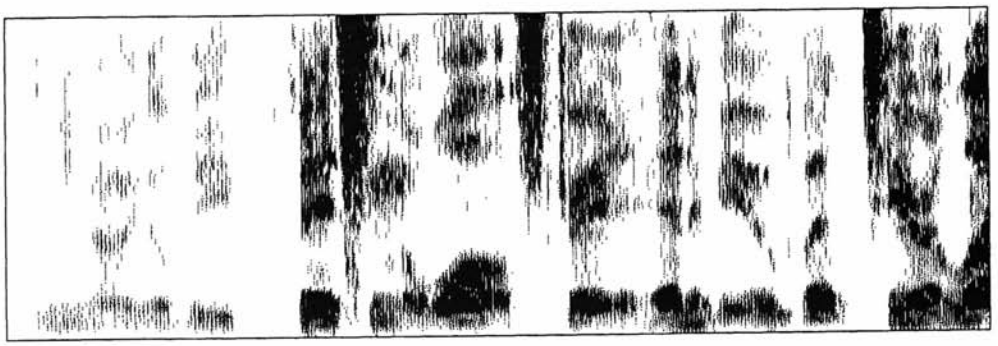


Figure 1.3. A spectrogram of 2.5 seconds of speech. The vertical axis is frequency, the horizontal axis is time, and energy maps to darkness.

speech for analysis by a human or a computer. People have been trained to read spectrograms and determine the words that were spoken. Although the spectrogram conveys salient features of a sound, the original acoustic signal cannot be reconstructed from it without some difficulty. As a result, the spectrogram is more useful for analysis of the speech signal than as a means of storing it for later playback.

Other acoustic representations in the frequency domain are even more succinct, though they are more difficult for a human to process visually than a spectrogram. **Linear Prediction Coefficients** and **Cepstral Analysis,** for example, are two such techniques that rely heavily on digital signal processing.[5] Both of these techniques reveal the resonances of the vocal tract and separate information about how the sound was produced at the noise source and how it was modified by various parts of the vocal tract. Because these two techniques extract salient information about how the sound was articulated they are frequently used as representations for computer analysis of speech.

PHONEMES AND SYLLABLES

Two representations of speech which are more closely related to its lexical structure are phonemes and syllables. Phonemes are important small units, several of which make up most syllables.

Phonemes

A **phoneme** is a unit of speech, the set of which defines all the sounds from which words can be constructed in a particular language. There is at least one pair of

[5]Linear prediction will be explained in Chapter 3. Cepstral analysis is beyond the scope of this book.

Ex.1018 / Page 38 of 343
AMAZON.COM, INC.

words in a language for which replacing one phoneme with another will change what is spoken into a different word. Of course, not every combination of phonemes results in a word; many combinations are nonsense.

For example, in English, the words "bit" and "bid" have different meanings, indicating that the "t" and "d" are different phonemes. Two words that differ in only a single phoneme are called a **minimal pair.** "Bit" and "bif" also vary by one sound, but this example does not prove that "t" and "f" are distinct phonemes as "bif" is not a word. But "tan" and "fan" are different words; this proves the phonemic difference between "t" and "f".

Vowels are also phonemes; the words "heed," "had," "hid," "hide," "howed," and "hood" each differ only by one sound, showing us that English has at least six vowel phonemes. It is simple to construct a minimal pair for any two vowels in English, while it may not be as simple to find a pair for two consonants.

An **allophone** is one of a number of different ways of pronouncing the same phoneme. Replacing one allophone of a phoneme with another does not change the meaning of a sentence, although the speaker will sound unnatural, stilted, or like a non-native. For example, consider the "t" sound in "sit" and "sitter." The "t" in "sit" is somewhat aspirated; a puff of air is released with the consonant. You can feel this if you put your hand in front of your mouth as you say the word. But in "sitter" the same phoneme is not aspirated; we say the aspiration is not phonemic for "t" and conclude that we have identified two allophones. If you aspirate the "t" in "sitter," it sounds somewhat forced but does not change the meaning of the word.

In contrast, aspiration of stop consonants is phonemic in Nepali. For an example of English phonemes that are allophones in another language, consider the difficulties Japanese speakers have distinguishing our "l" and "r." The reason is simply that while these are two phonemes in English, they are allophonic variants on the same phoneme in Japanese. Each language has its own set of phonemes and associated allophones. Sounds that are allophonic in one language may be phonemic in another and may not even exist in a third. When you learn a language, you learn its phonemes and how to employ the permissible allophonic variations on them. But learning phonemes is much more difficult as an adult than as a child.

Because phonemes are language specific, we can not rely on judgments based solely on our native languages to classify speech sounds. An individual speech sound is a **phone,** or **segment.** For any particular language, a given phoneme will have a set of allophones, each of which is a segment. Segments are properties of the human vocal mechanism, and phonemes are properties of languages. For most practical purposes, phone and phoneme may be considered to be synonyms.

Linguists use a notation for phones called the **International Phonetic Alphabet,** or **IPA.** IPA has a symbol for almost every possible phone; some of these symbols are shown in Figure 1.4. Since there are far more than 26 such phones, it is not possible to represent them all with the letters of the English alphabet. IPA borrows symbols from the Greek alphabet and elsewhere. For example, the "th" sound in "thin" is represented as "θ" in IPA, and the sound

Ex.1018 / Page 39 of 343
AMAZON.COM, INC.

Phoneme	Example Word	Phoneme	Example Word	Phoneme	Example Word
i	beet	p	put	č	chin
I	bit	t	tap	ǰ	judge
ε	bet	k	cat	m	map
e	bait	b	bit	n	nap
æ	bat	d	dill	η	sing
α	cot	g	get	r	ring
⊃	caught	f	fain	l	lip
∧	but	θ	thin	w	will
o	boat	s	sit	y	yell
U	foot	∫	shoe	h	head
u	boot	v	veal		
ɜ	bird	δ	then		
αj (αI)	bite	z	zeal		
⊃j (⊃I)	boy	ž	azure		
αw (αU)	bout				
ə	about				

Figure 1.4. The English phonemes in IPA, the International Phonetic Alphabet.

in "then" is "δ."[6] American linguists who use computers have developed the **Arpabet,** which uses ordinary alphabet characters to represent phones; some phonemes are represented as a pair of letters. Arpabet[7] was developed for the convenience of computer manipulation and representation of speech using ASCII-printable characters.

To avoid the necessity of the reader learning either IPA or Arpabet, this book indicates phones by example, such as, "the 't' in bottle." Although slightly awkward, such a notation suffices for the limited examples described. The reader will find it necessary to learn a notation (IPA is more common in textbooks) to make any serious study of phonetics or linguistics.

A phonemic transcription, although compact, has lost much of the original signal content, such as pitch, speed, and amplitude of speech. Phonemes are abstractions from the original signal that highlight the speech-specific aspects of that

[6]Are these sounds phonemic in English?

[7]The word comes from the acronym ARPA, the Advanced Research Projects Agency (sometimes called DARPA), a research branch of the U.S. Defense Department that funded much early speech research in this country and continues to be the most significant source of government support for such research.

Ex.1018 / Page 40 of 343
AMAZON.COM, INC.

signal; this makes a phonemic transcription a concise representation for lexical analysis as it is much more abstract than the original waveform.

Syllables

Another natural way to divide speech sounds is by the **syllable.** Almost any native speaker of English can break a word down into syllables, although some words can be more difficult, e.g., "chocolate" or "factory." A syllable consists of one or more consonants, a vowel (or diphthong[8]), followed by one or more consonants; consonants are optional, but the vowel is not. Two or more adjacent consonants are called a consonant **cluster;** examples are the initial sounds of "screw" and "sling." Acoustically, a syllable consists of a relatively high energy core (the vowel) optionally preceded or followed by periods of lower energy (consonants). Consonants have lower energy because they impose constrictions on the air flow from the lungs. Many natural languages, such as those written with the Arabic script and the northern Indian languages, are written with a syllabic system in which one symbol represents both a consonant and its associated vowel.

Other Representations

There are many other representations of speech appropriate to higher layer aspects of conversation.[9] Lexical analysis reveals an utterance as a series of **words.** A dictionary, or **lexicon** lists all the words of a language and their meanings. The **phrase,** sometimes called a "breath group" when describing intonation, is relevant both to the study of prosody (pitch, rhythm, and meter) as well as syntax, which deals with structures such as the noun phrase and verb phrase. A **parse tree,** as shown in Figure 1.5, is another useful representation of the syntactic relationships among words in a sentence.

Representations for higher layers of analysis are varied and complex. Semantics associates meaning with words, and meaning implies a relationship to other words or concepts. A **semantic network** indicates the logical relationships between words and meaning. For example, a door is a physical object, but it has specific meaning only in terms of other objects, such as walls and buildings, as it covers entrance holes in these objects.

Discourse analysis has produced a variety of models of the focus of a conversation. For example, one of these uses a **stack** to store potential topics of current focus. New topics are pushed onto the stack, and a former topic again becomes the focus when all topics above it are popped off the stack. Once removed from the stack, a topic cannot become the focus without being reintroduced.

[8]A diphthong consists of two vowel sounds spoken in sequence and is considered a single phoneme. The two vowel sounds in a diphthong cannot be separated into different syllables. The vowels in "hi" and "bay" are examples of diphthongs.

[9]Most of the speech representations mentioned in this section will be detailed in Chapter 9.

Ex.1018 / Page 41 of 343
AMAZON.COM, INC.

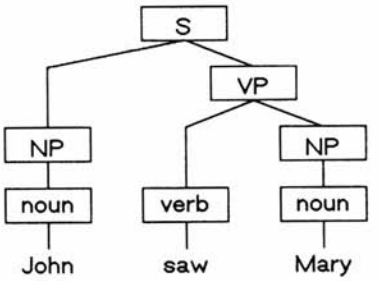


Figure 1.5. A parse tree.

SUMMARY

The perspective of speech as a conversational process constitutes the foundation and point of view of this book. Conversations employ spontaneous speech and a variety of interaction techniques to coordinate the exchange of utterances between participants. Speech is our primary medium of communication, although writing may be favored for longer-lasting or more formal messages. Computer systems would do well to exploit the richness and robustness of human conversation.

But "conversation" is simply too rich and varied a term to be amenable to analysis without being analyzed in components. This chapter identified a number of such components of a conversation: acoustic, articulatory, phonetic, lexical, syntactic, semantic, pragmatic, and discourse. Disciplines of research have been established for each of these areas, and the rest of this book will borrow heavily from them. Each discipline has its own set of representations of speech, which allow utterances to be described and analyzed.

Representations such as waveforms, spectrograms, and phonetic transcriptions provide suitable abstractions that can be embedded in the computer programs that attempt to implement various layers of speech communication. Each representation highlights particular features or characteristics of speech and may be far removed from the original speech sounds.

The rationale for this book is that the study of speech in conversations is interdisciplinary and that the designers of conversational computer systems need to understand each of these individual components in order to fully appreciate the whole. The rest of this book is organized in part as a bottom-up analysis of the layers of speech communication. Each layer interacts with the other layers, and the underlying goal for conversational communication is unification of the layers.

Ex.1018 / Page 42 of 343
AMAZON.COM, INC.

2

Speech Production and Perception

A basic knowledge of the physiology of speech production and its perception is necessary to understand both speech synthesis and speech recognition as well as compression techniques. This chapter provides an overview of the production and perception of speech. It briefly treats the organs of speech, i.e., the vocal tract and the auditory system. An articulatory model of speech is presented; this explains the different speech sounds of a language in terms of how they are produced. Finally, it introduces some basic findings of psychoacoustics that relate to the manner in which sound is perceived by the listener.

To distinguish the various types of sounds in English and the mechanisms whereby they are produced some signal processing terms are introduced. In this text, very simple and intuitive descriptions of such terms are given; more formal definitions may be found in texts on signal processing.

VOCAL TRACT

The vocal tract is the set of organs that produce speech sounds; these organs are also used for eating, drinking, and breathing. As seen in Figure 2.1, the vocal tract includes portions of the throat, mouth, and nasal cavities. These organs, the **articulators,** are moved to various configurations to produce the different sounds that constitute speech. The primary topic of this section is the production of speech sounds in general; the next section classifies the sounds specific to English.

Ex.1018 / Page 43 of 343
AMAZON.COM, INC.

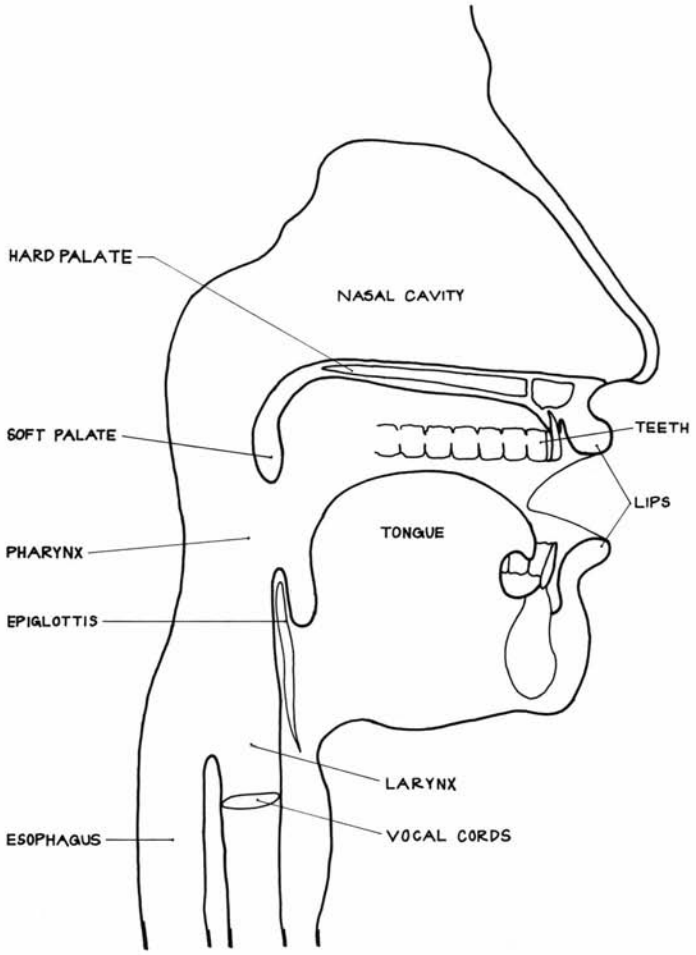


Figure 2.1. The major components of the vocal tract.

Sound is vibration in a medium, usually air, at frequencies that we can hear. In air, sound is carried as variations in pressure over time transmitted by the motion of air molecules. The air molecules create pressure variations by moving back and forth while not traveling a substantial distance. This variation is similar to the wave pattern generated when one end of a Slinky toy is moved back and forth. Some force must be created to cause this motion of molecules. In the vocal tract, this force is provided by air being expelled from the lungs by muscles in the diaphragm.

Without further modification, the air coming out of the lungs does not create a significant sound. If you open your mouth wide and exhale, you hear a little noise created as the air rushes out of the mouth, but this is usually not audible across a room. To become speech this air flow must be focused or concentrated in some

Ex.1018 / Page 44 of 343
AMAZON.COM, INC.

way. Focusing is accomplished by one of two mechanisms, either vibration of the vocal cords to produce a periodic sound or turbulent air flow through a constriction to produce an aperiodic, or "noisy," sound.[1]

If the vocal cords are vibrating, the speech produced is termed **voiced.** The vocal cords are folds of tissue capable of opening and closing in a regular fashion as controlled by several sets of muscles and driven by the pressure of air being expelled from the lungs. In operation the vocal cords remain closed until the pressure behind them is strong enough to force them open. As the air rushes by and pressure is released, a partial vacuum (Bernoulli effect) helps pull the cords back together again to shut off the flow of air.

This opening and closing pattern occurs at a regular rate and results in a series of pulses at that frequency. This frequency, called the **fundamental frequency** of voicing, or **F0,** corresponds to the pitch that we perceive in speech. F0 is typically higher for female and young speakers and may be varied within a limited range by muscle tension. The waveform of this glottal pulse is roughly triangular (see Figure 2.2), which results in a spectrum rich in harmonics, i.e., higher frequency components. The energy of speech drops at about 12 dB[2] per octave from the fundamental. Figure 2.3 illustrates the spectrum of the sound produced by the vocal chords. The **spectrum** is a measure of the magnitude of all the frequency components that constitute a signal. Since the source is of a single frequency F0, energy is found at that frequency and at integral multiples of that frequency. For example, the author's typical fundamental frequency is at 110 Hz,[3] which means that energy will be found at 220 Hz, 330 Hz, 440 Hz, etc.

You can experience voicing directly for yourself. Place a finger at the front bottom of your throat, slightly above the level of the shoulders. Speak out loud, alternating making "s" and "z" sounds. During the voiced "z" you can feel the vocal cords vibrate with your finger.

Another source of sound in the vocal tract is turbulence as air rushes through a constriction. Turbulence is not periodic; it is caused by the continuous stream of air molecules bouncing around randomly. It is similar in nature to the sound made by a babbling brook or the wind blowing through trees or surf moving onto shore. Such sounds contain a relatively large amount of high frequency energy, or "hiss," because the acoustic waveform varies a great deal from moment to moment. Turbulence is characteristic of both continuous **frication** sounds, such as the phoneme "s," and sudden **plosive** ones associated with a quick opening of a closed vocal tract, as in the phoneme "p." This turbulence can take place at a number of locations in the vocal tract. In the fricative "s," it is at a constriction formed between the tongue and the roof of the mouth. With the plosive "p," it is

[1]A periodic signal repeats a pattern at a regular interval or frequency. Aperiodic signals do not exhibit such patterns. Wind instruments produce periodic sounds, while percussion instruments are generally aperiodic.

[2]1 dB, short for decibel, corresponds to a just noticeable difference in sound energy.

[3]Hz, or Hertz, means cycles (or periods) per second. 1 kHz, or one kiloHertz, is one thousand cycles per second. 1 MHz, or MegaHertz, is one million cycles per second.

Ex.1018 / Page 45 of 343
AMAZON.COM, INC.

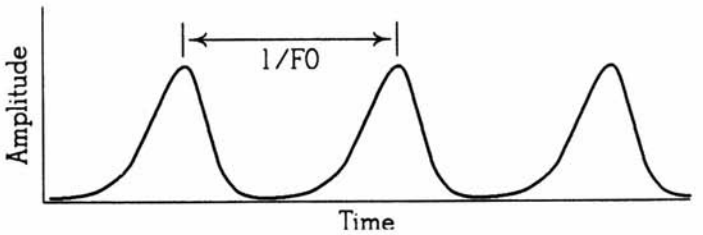


Figure 2.2. The glottal pulse is a roughly triangular-shaped waveform.

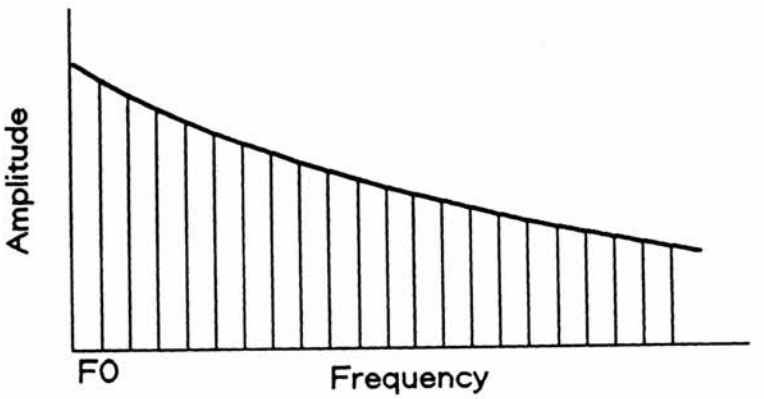


Figure 2.3. The spectrum of the sound produced by the vocal chords shows gradually decreasing magnitude with increasing frequency. This is properly a **line spectrum,** as energy is found only at multiples of F0, shown by the vertical line. The curve connecting the tops of these lines shows the **envelope,** or overall shape of the spectrum.

formed at the lips. The frication in "f" occurs around the lower lip making contact with the upper teeth.

There is more to classifying speech than identifying the source of the noise. The sound created by the noise source must then pass through the rest of the vocal tract that modifies it. These modifications are responsible for many of the distinguishing features of the different phonemes, or speech sounds. Even if the sound source is at the lips such that it does not have to pass through any further portions of the vocal tract, the size and shape of the oral cavity behind the lips affects the sound quality.

The vocal tract is usually analyzed as a series of **resonances.** A resonator is a physical system that enhances certain frequencies or range of frequencies. Resonators that amplify a very narrow range of frequencies are referred to as "high

Q",[4] and those which have a very broad band are "low Q" (see Figure 2.4). Because the vocal tract is composed of soft tissue, its resonators are rather low Q. In contrast, a very "clear" sound, such as a flute note, is produced by a high Q resonator.

The vocal tract is about 17 cm long and can be approximated as a series of joined tubes of varying diameters and lengths that add up to 17. One tube extends from the vocal cords up to the top of the mouth, the other, at approximately a right angle to the first, extends from there to the lips. The relative lengths of these tubes can be varied by the positioning of the tongue. If the tongue is low and flat (as in "ah"), then the front tube formed by the mouth is pronounced and the back tube is shorter. When the tongue is high in the mouth (as when saying the letter "e"), the front tube in front of the tongue is quite short, while the back tube is elongated.

If a column of air in a tube is excited by a source, it resonates at a characteristic frequency dependent upon the length of the tube. This is the principle upon which the flute is designed; opening the finger holes allows the air to escape at various points along the tube, thereby changing the length of the air column and thus its frequency. In the same way, the sound produced from the vocal cord vibration is enhanced at the frequencies corresponding to the length of the tubes in the vocal tract (see Figure 2.5). The frequencies at which these resonances occur are called **formants.** The lowest-frequency formant is labeled **F1** and called the first formant; the second formant is labeled **F2**, and so on. Most analysis stops at five or six formants. In the multitube model, each formant corresponds to a tube section.

Whichever the noise source, either voiced or turbulent, its spectrum is enhanced around the formant frequencies, which depend on the physical configuration of the vocal tract. Another part of the vocal tract that influences the noise source is the **nasal cavity,** a fairly soft region above the roof of the mouth and behind the nose. The passage from the back of the mouth to the nasal cavity can be closed by the **velum,** a small flap of tissue toward the top back of the mouth. When the velum is opened, the nasal cavity is physically connected and acoustically coupled with the vocal tract. The nasal cavity absorbs a significant amount of sound energy in the lower portion of the energy spectrum.

The radiation of the speech signal from the vocal tract into the surrounding air must also be considered. In addition to the mouth, sound is radiated from the cheeks, throat, and bony structure of the head and jaw, resulting in a signal that is much less directional than it would be otherwise with radiational patterns that are heavily frequency dependent. Flanagan [Flanagan 1960] measured sound energies of various frequencies at multiple locations about the head and found that at approximately 45 degrees (horizontally) away from the axis of the mouth

[4]Q, or Quality factor, is a measure of the difference between the higher and lower bounds of the frequencies amplified by the resonator, as measured at the point of half-power relative to the peak power at the center frequency. This number is divided by the center frequency; i.e., it is a factor of frequencies rather than a difference.

Ex.1018 / Page 47 of 343
AMAZON.COM, INC.

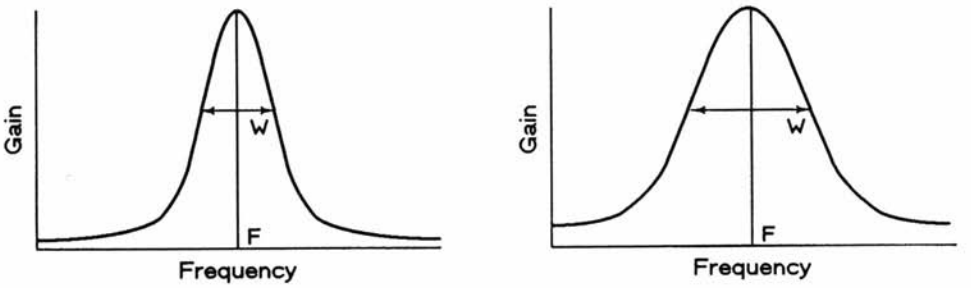


Figure 2.4. A high Q filter, left, and a low Q filter, right. F is the center frequency, and W is its bandwidth.

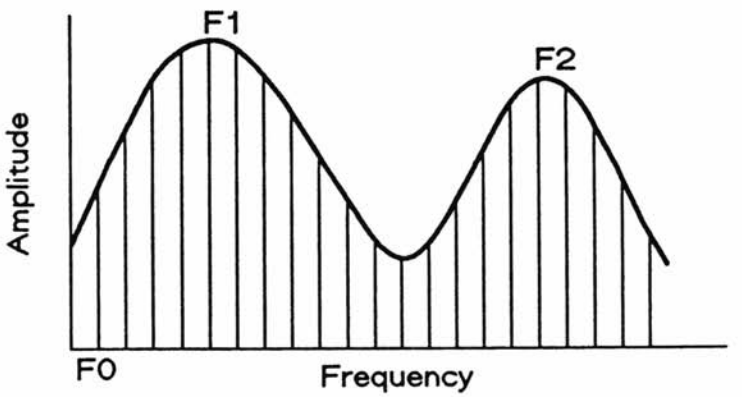


Figure 2.5. Resonances or formants of the vocal tract. The formants, labelled F1 and F2, are peaks in the vocal tract's filter function.

the signal was one half as strong, while directly behind the head the signal decreased by one order of magnitude less than in front of the head.

The noise source is referred to as the **source,** or **excitation,** signal. The manner in which the vocal tract enhances or diminishes the various frequencies of the signal as a function of its physical configuration is called its **transfer function.** A well-defined mathematical relationship allows computation of the signal that results from a source and a transfer function, but a detailed analysis of this relationship is beyond the scope of this book.

THE SPEECH SOUNDS

Each language employs a characteristic set of phonemes or allophones to convey words. These speech sounds are generated by manipulating the components of the vocal tract into specific physical configurations. The set of sounds comprising

Ex.1018 / Page 48 of 343
AMAZON.COM, INC.

the phonemes of a language must be distinctive because the listener perceives only the sound, not the vocal tract configuration that produces it. The classification of phonemes by the manner in which they are produced is referred to as an **articulatory model** of speech.

The three fundamental articulatory classifications are whether or not the phoneme is **voiced,** the **place** where the sound of the phoneme is made, and the **manner** in which it is made. The remainder of this section describes the various sounds of English according to their articulation.

Vowels

Vowels are sounds produced by vocal cord vibration (voicing) and a relatively open vocal tract, i.e., the lips, tongue, or teeth do not close the passageway. In the English alphabet there are only five letters we call vowels, but in spoken English there are closer to 17 vowel phonemes. Vowels are steady state; after we begin speaking a vowel, we can continue it until we run out of breath. Because vowels are voiced, they are periodic as can be seen in Figure 2.6.

The vowels can be differentiated by the acoustical effect of the position of the tongue and lips during their pronunciation. We can distinguish three degrees of constriction of the vocal tract that indicate how close to the roof of the mouth the hump of the tongue gets (high, medium, and low). We can also differentiate three positions laterally where the hump of the tongue makes the greatest constriction (front, central, and back). In addition, vowels can be *nasalized* by moving the velum to open the air passage leading from the throat to the nose. English does not employ nasalization as a phonemic distinction; however, certain other languages such as French use nasalization as a distinguishing feature to contrast phonemes.

In English the vowels are distinguished solely by the position of the tongue, which changes the lengths of the two primary vocal tract tube sections. In other words, the first two formants form a set of distinctive features for the vowels. Although the typical positions of the formants will vary with gender and among speakers, they can generally be clustered together into regions in a space defined

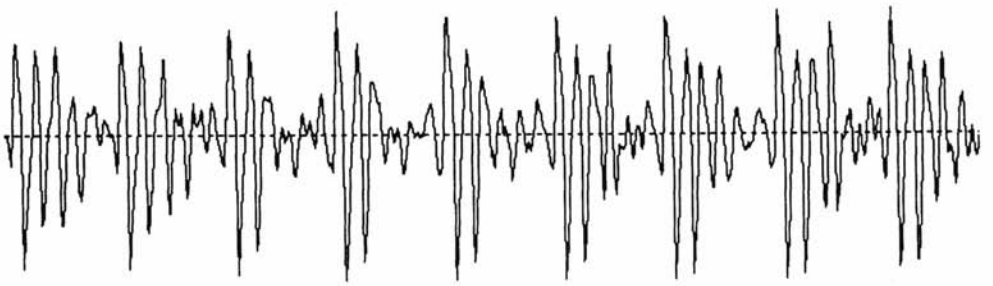


Figure 2.6. A waveform of the vowel in "but." The display shows amplitude mapped against time, over a 100 millisecond interval. Nine pitch periods can be seen; the frequency of voicing is about 90 Hz.

Ex.1018 / Page 49 of 343
AMAZON.COM, INC.

by F1 and F2 as illustrated in Figure 2.7. Such a diagram is sometimes called the "vowel triangle" because all of the vowels fit in a space that approximates a triangle in the chosen coordinate system of F1 versus F2.

Some vowels are **diphthongs.** Diphthongs are actually combinations of two vowel sounds with the vocal tract moving from one position to another during their production. The vowel sounds in "bay" and "boy" are examples of diphthongs.

Consonants

Consonants are characterized by constrictions in the vocal tract and can be differentiated by place of closure, manner (type or degree) of closure, and whether they are voiced.

Place refers to the location in the vocal tract of the closure associated with a consonant. **Labial** closure uses both lips as in "p" and "b" sounds. **Labial-dental** closure involves the lower lip and upper teeth as in "f" and "v" sounds. **Alveolar** closure involves the tongue and the gum ridge behind the front teeth as in "n," "d," "s," and "z" sounds. **Palatal** closure uses the tongue on the soft palate or roof of the mouth slightly farther back as in "sh" and "zh" sounds.

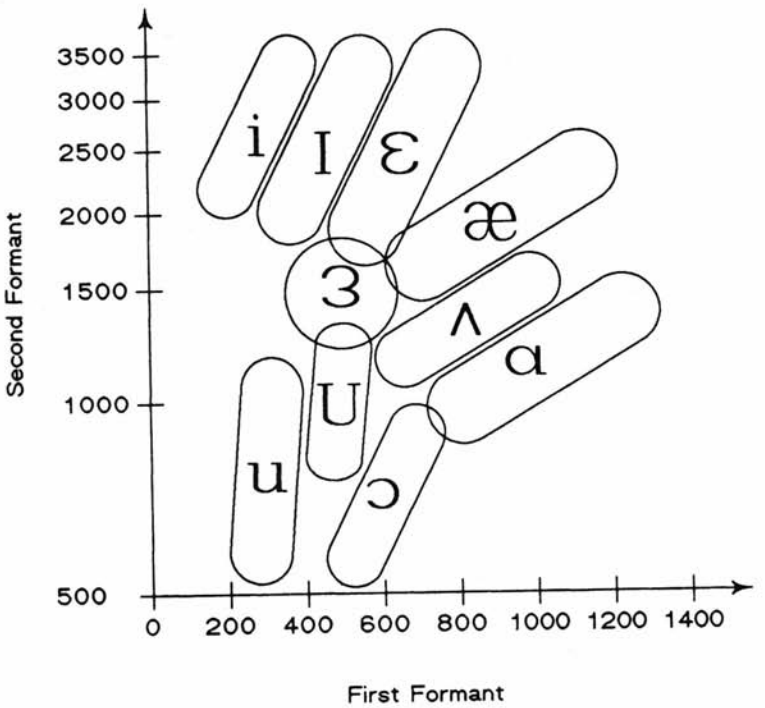


Figure 2.7. The vowel triangle displays English vowels mapped onto a Cartesian space defined by the the first and second formants. The IPA notation from Figure 1.4 is used for labeling.

Ex.1018 / Page 50 of 343
AMAZON.COM, INC.

Velar closure occurs at the back of the mouth, or hard palate, as in "g" and "k" sounds. Each of these places of closure results in a distinct sound as the shape of the vocal tract changes and consequently the frequencies of resonance vary. The closure divides the vocal tract into cavities in front of and behind the closure; these have different resonant frequencies depending on their size. For example, the position of the third formant is often a strong cue to the identity of a stop consonant.

Closure of the vocal tract is achieved in several different manners, each resulting in a different sound. **Stops** ("t," "b," "p," "d") involve a sudden and total cessation of air flow. Stop consonants are very dynamic. During the closure, no air flows through the vocal tract, resulting in silence. Noise resumes suddenly as the closure is released, and the air pressure from behind the closure may result in **aspiration** noise, such as in the "t" in "top." Stops are also called **plosives,** which focuses on the release rather than the closure of the consonant.

Fricatives ("s," "z," "sh," "zh") involve constriction to the point of producing turbulence and hence noise but not total closure. The partial closure for "s" is alveolar and that for "sh" is palatal. Because closure is incomplete the fricatives result in a continuous sound as air flows by. Sound created by air rushing through a small opening is aperiodic (see Figure 2.8) and dominated by high-frequency components. Combining these two traits, the characteristic acoustic feature of the fricatives is a moderate duration of high-frequency energy.

Nasals ("m," "n," "ng" as in "sing") are produced by closing the oral cavity but opening the velum to the nasal cavity. The nasal cavity absorbs a significant amount of low-frequency energy giving a distinctive cue to nasalization. The three English nasals are all voiced. The nasals are differentiated from each other by the place at which the vocal tract is closed, which is either labial ("m"), palatal ("n"), or velar ("ng").

Most of the consonant types mentioned so far have come in pairs, such as ("d," "t") and ("v," "f"). Both members of each of these pairs is produced with the articulators in the same place but are distinguished by the presence or absence of

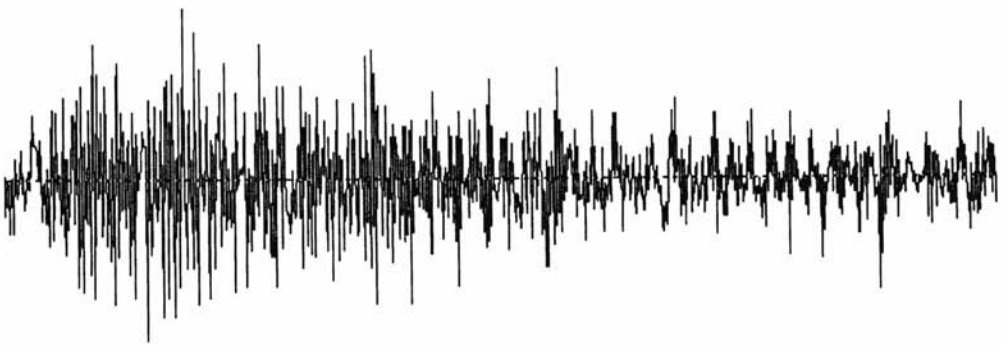


Figure 2.8. The aperiodic waveform of the consonant in "see." This figure also shows 100 milliseconds of speech. No periodicity is evident.

Ex.1018 / Page 51 of 343
AMAZON.COM, INC.

voicing. Voiced consonants thus have two noise sources: one at the place of articulation and the other at the periodic glottal pulses.

Whether or not a consonant is voiced depends on the type of consonant. In a stop there is no air flow so a stop is considered voiced if voicing resumes soon after the release; otherwise it is unvoiced. Air flow continues during a fricative so voicing is defined by whether the vocal cords are vibrating during the consonant itself.

Liquids and Glides

A small class of special-case phonemes behave much like vowels. The **liquids** ("l," "r") invariably precede a vowel and are dynamic, involving a transition into the vowel. It is sometimes hard to identify "r" as a distinct consonant but rather to observe the effect, or **coloration,** of the vowel caused by the motion of the lips and the rise of the tongue in the middle of the mouth. The **glides** ("w," "y" as in "you") are similar to vowels except that the vocal tract is more constricted than for the vowels. According to another point of view, the glides are simply vowels that do not occur during the energy peak of the syllable.

Acoustic Features of Phonemes

The methods used to articulate the various phonemes of a language must produce sounds with adequate acoustical cues to allow the listener to distinguish between possible phonemes in speech. Some examples of acoustical cues have just been mentioned: the distinctive structure of the first two formants in vowels, the energy at high frequencies that distinguish fricatives, the presence or absence of periodicity due to voicing, and the absorbing effect of the nasal cavity.

The listener can select from a multitude of cues occurring simultaneously to differentiate phonemes. In fluent speech the articulators often move only partially in the direction of target vocal tract configurations described above for each phoneme. The acoustical effect is still sufficiently pronounced to allow the listener to detect at least some of the cues and identify each phoneme.

HEARING

Sound arrives at our ears as variations in air pressure. We can hear vibrations in the range of approximately 20 Hz up to 15 or 20 kHz; this figure varies among individuals, and the range decreases as we age. This sensory stimulus triggers neural activity through a complex but versatile mechanical transformation in the ear. There is a chain of processes whereby the physical sound from its source causes the auditory event of "hearing." In turn, these neural firings stimulate perceptual processing at higher levels of the brain. Although little is known about such processing from direct neural evidence, the domain of **psychoacoustics** studies our perception of sound by observing subjects' responses to acoustic stimuli.

Ex.1018 / Page 52 of 343
AMAZON.COM, INC.

Auditory System

The complement to the vocal tract is the auditory system. Although primarily designed to turn air pressure variations into corresponding neural signals, the ear also contains the **vestibular organs** that are used to maintain physical balance; we are not concerned with them here. The ear can be divided into three distinct sections. The outer ear directs sound toward the eardrum, the middle ear converts the pressure variations of sound into mechanical motion, and the inner ear converts this motion to electrical signals in the auditory neurons. The ear is shown in Figure 2.9.

The **outer ear** includes the **pinna** and the **ear canal,** which leads to the eardrum. The pinna consists of the fleshy protrusion on the side of the head and is what we usually refer to when we use the term "ear" (as in "Van Gogh cut off his ear"). The pinna with its various folds of cartilage around the ear opening serves primarily as a protective mechanism. The pinna provides some amplification of the sound by focusing it into the ear canal in much the same way that we may cup our hands behind our ears to better hear quiet sounds. It is directional at high frequencies and is used as a localization aid to find a sound source because it makes the ear more sensitive to sounds coming from in front of rather than behind the listener. In certain animals such as owls, the outer ear occupies a much larger surface area with respect to the head and is more fundamental to localization.

The ear canal is a tube about 1 cm wide by 2.5 cm long leading to the middle ear. It has a resonance of about 3000 Hz and therefore amplifies sound at this frequency. The length of the ear canal shields the middle ear from physical injury if the side of the head is struck.

The **middle ear** provides the linkage between the outer and inner ear; its function is to effectively convert variations in air pressure to mechanical motion of the liquid inside the inner ear. The **eardrum,** or **tympanic membrane,** covers the interior end of the ear canal. Vibrations in the air cause the eardrum to vibrate; thus it produces physical motion from the sound in the air. The middle ear is filled with air. The **eustachean tube** runs between the middle ear and the throat. When we yawn, the eustachean tube opens, allowing the air pressure to equalize across the eardrum. As experienced when descending rapidly in an airplane, this pressure balance is essential for effective transfer of sound across the eardrum.

The inner ear is filled with water. There is a large impedance mismatch between air- and water-like fluids. Water is much denser than air so air does not cause significant displacement in the liquid that it impinges; even a strong wind on a pond does not penetrate much deeper than to form surface ripples because of this mismatch. A series of three **ossicular bones** (the **malleus, incus,** and **stapes**) provide mechanical advantage (leverage) from the ear drum to the **oval window,** a membrane on the surface of the inner ear. As the eardrum is 17 times as large as the oval window, this difference in area provides further amplification. The size difference plus the mechanical advantage of the ossicular bones combine to provide a 22:1 amplification and thereby an impedance match for efficient transfer of acoustical energy to the inner ear.

Ex.1018 / Page 53 of 343
AMAZON.COM, INC.

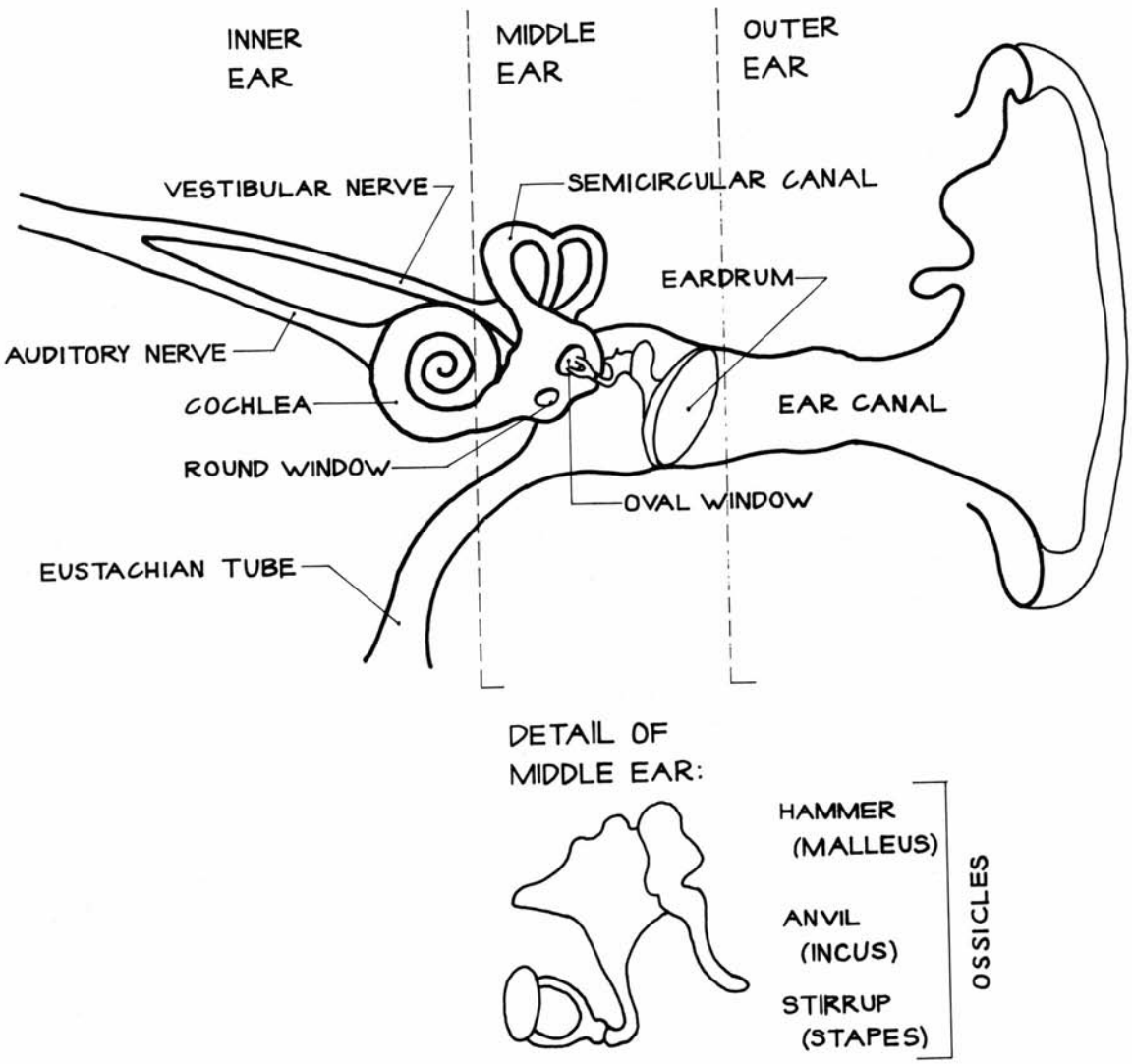


Figure 2.9. The ear.

The middle ear has several different modes of vibration that provide improved dynamic range without distortion (the goal of all good audio amplifiers). For loud sounds, the vibration mode is changed from pumping to a rotational action. In addition, for very loud sounds a reflex occurs in the muscles of the middle ear to damp the vibration and thereby protect the sensitive inner ear.

Ex.1018 / Page 54 of 343
AMAZON.COM, INC.

The **inner ear** is the transducer from mechanical to electrical energy. At its center is the **cochlea,** a spiral-shaped chamber looking somewhat like a snail shell and making 2⅝ turns. Sound-induced waves in the liquid medium of the cochlea cause vibration on the **basilar membrane.** Different portions of this tapering membrane vibrate in response to sound energy at specific frequencies. At the basal end it is thin and stiff, while at its apex it is flexible and massive.

The frequency dependent vibration of the basilar membrane induces motion in the microscopic hairs that penetrate the membrane. These hairs are linked to neurons to the brain that produce neural firings in response to the stimulus of the hairs' bending. Each neuron is connected to a small number of hairs in a particular location on the basilar membrane, and the basilar membrane responds to different frequencies along its length so the firing of a neuron corresponds to the presence of sound of the appropriate frequency. Because of this relationship between neural firings and basilar membrane vibration, an individual neuron is not very responsive to frequencies much higher or lower than its preferred one. Measurements of neural responses indicate that each acts as a bandpass filter and that all the neurons and associated hair cells exhibit a fairly constant Q.

Neural activity is highest at the onset of a sound and decays rapidly at first and then more slowly. This is known as **adaptation;** we are more sensitive to changing sounds than continual tones. Neurons fire more rapidly for louder sounds, reaching peak rates of up to 1000 firings per second for short periods. At frequencies below about 1 kHz, neurons tend to fire in phase with the vibration of the basilar membrane, i.e., every cycle of membrane motion induces one neural spike.

The ear's response to a sound is a series of neural spikes from many neurons simultaneously. The firing pattern of any particular neuron is a function of amplitude of the acoustic energy (at the basilar membrane) within the frequency range to which that neuron is sensitive. The pattern adapts over a short period of time and may be at least partially in phase with the acoustic signal. The neural firings are transmitted to a number of stages of processing in the central auditory system. At several points along the way are sites that are particularly sensitive to time or amplitude differences between signals arriving from each ear. Such differences are central to our ability to locate a sound spatially.

Localization of Sounds

When we hear a sound, we perceive it as coming from some location in space outside of our head; this is known as **localization.** Much of our ability to localize sound depends on the differences in the sound that arrives at each of our two ears from a single source. These differences are due to the positions of the ears and hence their different distances from the sound source as well as the tendency of the head to partially block sound coming to an ear from the opposite side of the head.

We can localize a sound source within about 4 degrees in front of the head, 6 degrees behind the head, and only within 10 degrees at the sides. Localization is very frequency dependent and we are most sensitive to sounds at around 800 Hz.

Ex.1018 / Page 55 of 343
AMAZON.COM, INC.

Our hearing is about 100 times less sensitive to position than our visual system, but we can hear sounds behind our heads. We have a strong reflex to turn and look at a sound that occurs behind our heads. This moves the source of the sound into our field of view, which could aid survival, and places the source where we can better localize it. The motion of the head also provides cues to enhance localization as we seem to be especially sensitive to differences in location.

If a sound source is off to either side of our head, it takes the sound longer to reach the ear on the other side as it is further away (see Figure 2.10). In other words, when a sound begins one ear hears it before the other. But more importantly, while the sound continues the air vibrations will arrive at each ear out of phase with the other ear; this is defined as the **interaural phase difference.** The human head has a thickness of 21 or 22 centimeters, which means that it can take sound up to about 600 microseconds longer to reach the further ear. This phase difference becomes confused when it is greater than the wavelength of the sound so phase is most effective for localizing sound below 1500 Hz.

When a sound is off to one side, the mass of the head also blocks the direct path from the sound source to the opposite ear. This results in an **interaural intensity difference;** the sound is louder at the ear that has an unobscured path. This effect is most pronounced at frequencies above 10 kHz, as the head more effectively blocks small wavelengths.

The differences in phase and intensity of the sound arriving at each ear can help localize it to the left or right side but leave confusion as to whether the sound is in front of or behind the head. A sound in front of and to the left of the listener creates the same interaural phase and intensity differences as a sound the same distance behind and also to the left. The shape of the pinna interacts with incom-

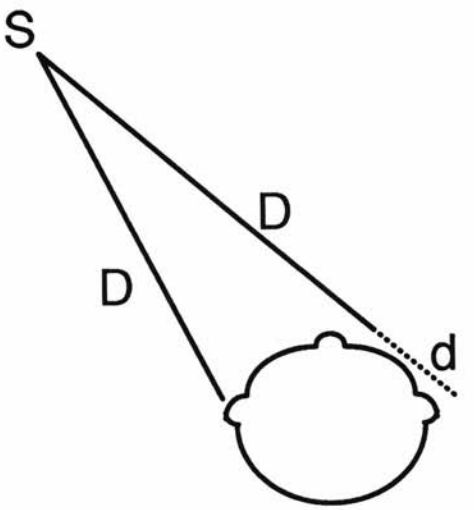


Figure 2.10. A sound on the left side of the head arrives at the right ear later than the left ear because it must travel further.

Ex.1018 / Page 56 of 343
AMAZON.COM, INC.

ing sound and provides front-back cues. Its amplification effect provides directionality through increased gain for sounds from the front. In addition, the folds of the pinna act as a complex filter enhancing some frequencies; this filtering is different for sounds from the front than from the back providing further front/back cues.

Psychoacoustics

So far we have described the process whereby sound energy is converted to neural signals; we have not yet described how the listener perceives the sound characterized by such signals. Although more is known about higher-level processing of sound in the brain than we have mentioned in the brief overview above, there is still not sufficient knowledge to describe a model of such higher auditory processing. However, researchers can build an understanding of how we perceive sounds by presenting acoustical stimuli to subjects and observing their responses. The field of psychoacoustics attempts to quantify perception of pitch, loudness, and position of a sound source.

The ear responds to a wide but limited range of sound loudness. Below a certain level, sound cannot be perceived; above a much higher level, sound causes pain and eventually damage to the ear. Loudness is measured as **sound pressure level** (SPL) in units of **decibels** (dB). The decibel is a logarithmic scale; for power or energy, a signal that is X times greater than another is represented as $10 \log_{10} X$ dB. A sound twice as loud is 3 dB higher, a sound 10 times as loud is 10 dB higher, and a sound 100 times as loud is 20 dB higher than the other. The reference level for SPL is 0.0002 dyne/cm^2; this corresponds to 0 dB.

The ear is sensitive to frequencies of less than 20 Hz to approximately 18 kHz with variations in individuals and decreased sensitivity with age. We are most sensitive to frequencies in the range of 1 kHz to 5 kHz. At these frequencies, the threshold of hearing is 0 dB and the threshold of physical feeling in the ear is about 120 dB with a comfortable hearing range of about 100 dB. In other words, the loudest sounds that we hear may have ten billion times the energy as the quietest sounds. Most speech that we listen to is in the range of 30 to 80 dB SPL.

Perceived loudness is rather nonlinear with respect to frequency. A relationship known as the Fletcher-Munson curves [Fletcher and Munson 1933] maps equal input energy against frequency in equal loudness contours. These curves are rather flat around 1 kHz (which is usually used as a reference frequency); in this range perceived loudness is independent of frequency. Our sensitivity drops quickly below 500 Hz and above about 3000 Hz; thus it is natural for most speech information contributing to intelligibility to be within this frequency range.

The temporal resolution of our hearing is crucial to understanding speech perception. Brief sounds must be separated by several milliseconds in order to be distinguished, but in order to discern their order about 17 milliseconds difference is necessary. Due to the firing patterns of neurons at initial excitation, detecting details of multiple acoustic events requires about 20 milliseconds of averaging. To note the order in a series of short sounds, each sound must be between 100 and 200 milliseconds long, although this number decreases when the sounds have

Ex.1018 / Page 57 of 343
AMAZON.COM, INC.

gradual onsets and offsets.[5] The prevalence of such transitions in speech may explain how we can perceive 10 to 15 phonemes per second in ordinary speech.

Pitch is the perceived frequency of a sound, which usually corresponds closely to the fundamental frequency, or F0, of a sound with a simple periodic excitation. We are sensitive to *relative* differences of pitch rather than absolute differences. A 5 Hz difference is much more significant with respect to a 100 Hz signal, than to a 1 kHz signal. An **octave** is a doubling of frequency. We are equally sensitive to octave differences in frequencies regardless of the absolute frequency; this means that the perceptual difference between 100 Hz and 200 Hz is the same as between 200 Hz and 400 Hz or between 400 Hz and 800 Hz.

The occurrence of a second sound interfering with an initial sound is known as **masking.** When listening to a steady sound at a particular frequency, the listener is not able to perceive the addition of a lower energy second sound at nearly the same frequency. These frequencies need not be identical; a sound of a given frequency interferes with sounds of similar frequencies over a range called a **critical band.** Measuring critical bands reveals how our sensitivity to pitch varies over the range of perceptible frequencies. A pitch scale of **barks** measures frequencies in units of critical bands. A similar scale, the **mel** scale, is approximately linear below 1 kHz and logarithmic above this.

Sounds need not be presented simultaneously to exhibit masking behavior. **Temporal masking** occurs when a sound masks the sound succeeding it (**forward** masking) or the sound preceding it (**backward** masking). Masking influences the perception of phonemes in a sequence as a large amount of energy in a particular frequency band of one phoneme will mask perception of energy in that band for the neighboring phonemes.

SUMMARY

This chapter has introduced the basic mechanism for production and perception of speech sounds. Speech is produced by sound originating at various locations in the vocal tract and is modified by the configuration of the remainder of the vocal tract through which the sound is transmitted. The articulators, the organs of speech, move so as to make the range of sounds reflected by the various classes of phonemes.

Vowels are steady-state sounds produced by an unobstructed vocal tract and vibrating vocal cords. They can be distinguished by their first two resonances, or formants, the frequencies of which are determined primarily by the position of the tongue. Consonants are usually more dynamic; they can be characterized according to the place and manner of articulation. The positions of the articula-

[5]The onset is the time during which the sound is just starting up from silence to its steady state form. The offset is the converse phenomenon as the sound trails off to silence. Many sounds, especially those of speech, start up and later decay over at least several pitch periods.

Ex.1018 / Page 58 of 343
AMAZON.COM, INC.

tors during consonant production create spectra characteristic of each phoneme; each phoneme has multiple cues as to its identity.

Sound consists of vibrations in a medium (usually air) at a frequency we can hear. The variations in sound pressure are focused by the outer ear and cause mechanical motion in the middle ear. This mechanical motion results in waves in the liquid in the inner ear, which causes neurons to fire in response. Which neurons fire and the pattern of their firings is dependent on the spectral and temporal characteristics of the sound. We localize a sound based on interaural phase and intensity differences. Each ear receives a slightly different sound signal due to the position of the ears on either side of the head. The pinna enables us to distinguish between sounds in front and behind.

This chapter also discussed several topics in psychoacoustics and the frequency and temporal responses of the auditory system; what we perceive with our sense of hearing is as much due to the characteristics of our auditory system as to the qualities of the sounds themselves. These factors influence our perception of speech in particular as well as all sounds.

The next chapter explores methods to digitally capture, encode, and compress the speech signal for computer storage, analysis, or transmission. Data compression techniques for speech take advantage of temporal and frequency characteristics of the speech signal as well as the sensitivity of the auditory system to reduce the data rate without loss of intelligibility or perceived quality of the reconstructed signal.

FURTHER READING

A concise overview of the human speech system can be found in Denes and Pinson and a more analytic and rigorous one in Flanagan (1972). Ladefoged offers an excellent treatment of phonetics. O'Shaughnessy covers many issues in speech production and hearing as a prelude to describing speech technologies. Handel offers an excellent overview of the perception of speech and music, while Yost is a more introductory text with superb illustrations.

Ex.1018 / Page 59 of 343
AMAZON.COM, INC.

3

Speech Coding

This chapter introduces the methods of encoding speech digitally for use in such diverse environments as talking toys, compact audio discs, and transmission over the telephone network. To use voice as data in such diverse applications as voice mail, annotation of a text document, or a talking telephone book, it is necessary to store speech in a form that enables computer retrieval. Digital representations of speech also provide the basis for both speech synthesis and voice recognition. Chapter 4 discusses the management of stored voice for computer applications, and Chapter 6 explore interface issues for interactive voice response systems.

Although the use of a conventional analog tape recorder for voice storage is an option in consumer products such as answering machines, its use proves impractical in more interactive applications. Tape is serial, requiring repeated use of fast-forward and rewind controls to access the data. The tape recorder and the tape itself are prone to mechanical breakdown; standard audiotape is unable to endure the cutting and splicing of editing.

Alternatively, various digital coding schemes enable a sound to be manipulated in computer memory and stored on random access magnetic disks. Speech coding algorithms provide a number of digital representations of a segment of speech. The goal of these algorithms is to faithfully capture the voice signal while utilizing as few bits as possible. Minimizing the data rate conserves disk storage space and reduces the bandwidth required to transmit speech over a network; both storage and bandwidth may be scarce resources and hence costly. A wide range of speech quality is available using different encoding algorithms; quality requirements vary from application to application. In general, speech quality may be traded for information rate by choosing an appropriate coding algorithm.

Ex.1018 / Page 60 of 343
AMAZON.COM, INC.

Some formerly popular coding schemes, such as Linear Predictive Coding, made sizable sacrifices of quality for a low bit rate. While this may have been dictated by the technology of the time, continually decreasing storage costs and increasingly available network bandwidth now allow better-quality speech coding for workstation environments. The poorer-quality encoders do find use, however, in military environments where encryption or low transmitter power may be desired and in consumer applications such as toys where reasonable costs are essential.

The central issue in this chapter is the proposed trade-off between bit rate and intelligibility; each generation of speech coder attempts to improve speech quality and decrease data rate. However, speech coders may incur costs other than limited intelligibility. Coder complexity may be a factor; many coding algorithms require specialized hardware or digital signal processing microprocessors to run in real time.[1] Additionally, if the coder uses knowledge of the history of the signal to reduce the data rate, some flexibility may be lost in the encoding scheme, making it difficult both to edit the stored speech and to evaluate the encoded signal at a particular moment in time. And finally, although it is of less concern here, coder robustness may be an issue; some of the algorithms fail less gracefully, with increased sound degradation if the input data is corrupted.

After discussing the underlying process of digital sampling of speech, this chapter introduces some of the more popular speech coding algorithms. These algorithms may take advantage of a priori knowledge about the characteristics of the speech signal to encode only the most perceptually significant portions of the signal, reducing the data rate. These algorithms fall into two general classes: **waveform coders,** which take advantage of knowledge of the speech signal itself and **source coders,** which model the signal in terms of the vocal tract.

There are many more speech coding schemes than can be detailed in a single chapter of this book. The reader is referred to [Flanagan et al. 1979, Rabiner and Schafer 1978] for further discussion. The encoding schemes mentioned here are currently among the most common in actual use.

SAMPLING AND QUANTIZATION

Speech is a continuous, time-varying signal. As it radiates from the head and travels through the air, this signal consists of variations in air pressure. A microphone is used as a transducer to convert this variation in air pressure to a corresponding variation in voltage in an electrical circuit. The resulting electrical signal is analog. An analog signal varies smoothly over time and at any moment is one of an unlimited number of possible values within the dynamic range of the

[1]A real-time algorithm can encode and decode speech as quickly as it is produced. A non-real-time algorithm cannot keep up with the input signal; it may require many seconds to encode each second of speech.

Ex.1018 / Page 61 of 343
AMAZON.COM, INC.

medium in which the signal is transmitted.[2] In analog form, the signal may be transmitted over a telephone circuit (as voltage) or stored on a magnetic tape (as magnetic flux).

All the sensory stimuli we encounter in the physical world are analog. For an analog signal to be stored or processed by a computer, it must first be transformed into a digital signal, i.e., **digitized.** Digitization is the initial step of signal processing for speech recognition. Speech synthesis algorithms also generate digital signals, which are then transformed into analog form so we can hear them.

A digitized signal consists of a series of numeric values at some regular rate (the sampling rate). The signal's possible values are constrained by the number of bits available to represent each sampled value—this simple representation is called **Pulse Code Modulation (PCM).** To convert an analog signal to this digital form it must be **sampled** and **quantized.**

Sampling assigns a numeric value to the signal in discrete units of time, i.e., a new value is given for the signal at a constant rate. The Nyquist rate specifies the maximum frequency of the signal that may be fully reproduced: a signal must be sampled at twice its highest frequency to be faithfully captured [Nyquist 1924]. In addition, when the analog signal is being recreated, it must be passed through a low-pass filter at one-half the sampling rate to remove audible artifacts of the sampling frequency.[3]

If these sampling rate conditions are not met, **aliasing** results. Aliasing is a form of distortion in which the high frequencies are falsely regenerated as lower frequencies. A rigorous description of aliasing requires mathematics beyond the scope of this book; nonetheless, the following example offers a degree of explanation. While watching a western film, you may have noticed that the wagon wheels sometimes appear to rotate backwards or in the correct direction but at the wrong rate; the same phenomenon occurs with airplane propellers, fans, etc. This is an example of aliasing. In actuality, the wheel is turning continuously, but film captures images at a rate of 24 frames per second. A repetitive motion that occurs at more than half this sampling rate cannot be captured correctly; when we view the film we perceive a motion that may be faster, slower, or of opposite direction.

Why does this happen? Consider the eight-spoked wagon wheel shown in Figure 3.1; as it rotates clockwise, the spokes move together. If the spokes move just a little in the 1/24 of a second between frames (see Figure 3.2), they will appear to be moving forward at the correct speed. If the wheel moves exactly 1/8 of a revolution between frames, the spokes will have changed position (see Figure 3.3); but since the spokes all look identical, the film audience will not see the spokes rotating even though they have moved forward. This is, of course, an illusion, as the stagecoach would be moving against the background scenery. If the wheel rotates slightly faster such that each spoke has traveled beyond the position originally

[2]An electric circuit saturates at a maximum voltage, and below a minimum level the voltage variations are not detectable.

[3]In practice, to ensure that the first condition is actually met during sampling, a similar filter at one half the sampling frequency is used at the input stage.

Ex.1018 / Page 62 of 343
AMAZON.COM, INC.

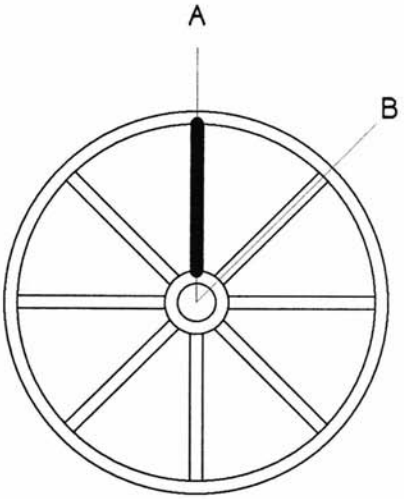


Figure 3.1. A wagon wheel with two spokes marked for reference points.

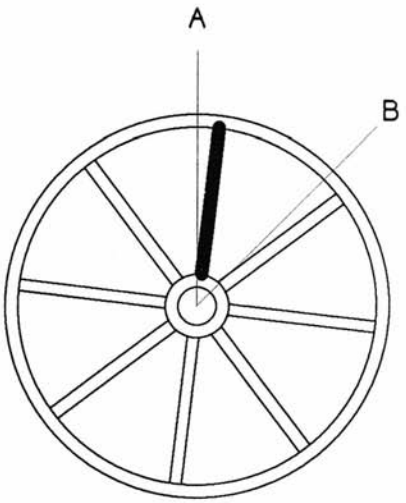


Figure 3.2. The wheel has rotated forward a small distance.

occupied by the next spoke (see Figure 3.4), the viewer will see a much slower motion identical to that depicted by Figure 3.2. Finally, if a spoke has moved most of the way to the next spoke position between frames (see Figure 3.5), it will appear that the wheel is slowly moving counterclockwise.

Except for the first case, these illusions are all examples of aliasing. The repetitive motion consists of identical spokes rotating into positions occupied by other

Ex.1018 / Page 63 of 343
AMAZON.COM, INC.

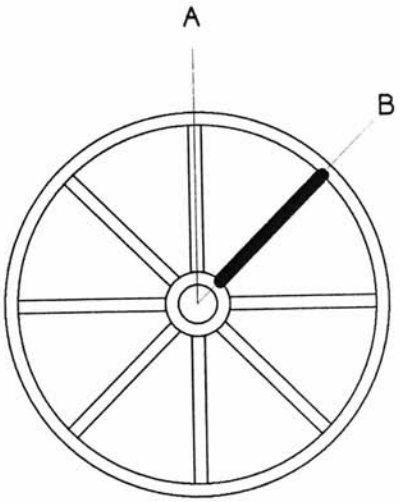


Figure 3.3. The wheel has rotated far enough that a spoke has moved forward to the position previously occupied by the next spoke.

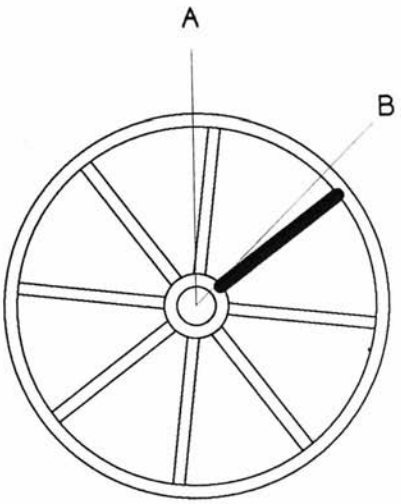


Figure 3.4. The wheel has rotated still further. Each spoke has moved just beyond the position formerly occupied by the next.

spokes in the time between frames. If aliasing is to be avoided, the sampling theorem suggests that this must happen at no more than one-half the sampling rate. This implies that each spoke must rotate no more than one-half the distance to the next spoke between frames; with eight spokes filmed at 24 frames per second, this corresponds to 1½ wheel revolutions per second. A characteristic

Ex.1018 / Page 64 of 343
AMAZON.COM, INC.

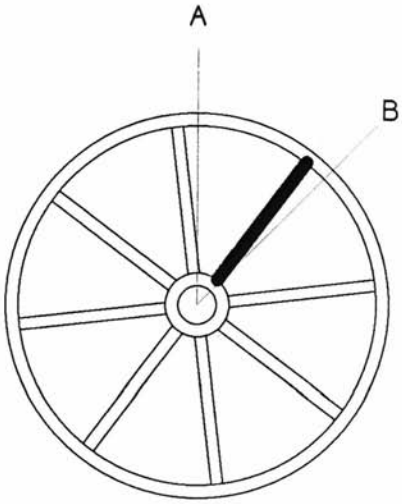


Figure 3.5. The spokes have moved almost to the position of the next spoke.

of aliasing is that frequencies above one-half the sampling rate appear incorrectly as lower frequencies. Identical aliasing artifacts also appear at multiples of the sampling frequency. The visual effects of the wheel moving a small fraction of the way to the next spoke, the distance to the next spoke plus the same small fraction, or the distance to the second spoke beyond plus the small fraction are all identical.

Another example provides a better understanding of sampling applied to an audio signal. Consider a relatively slowly varying signal as shown in Figure 3.6. If sampled at some rate **T,** a series of values that adequately capture the original

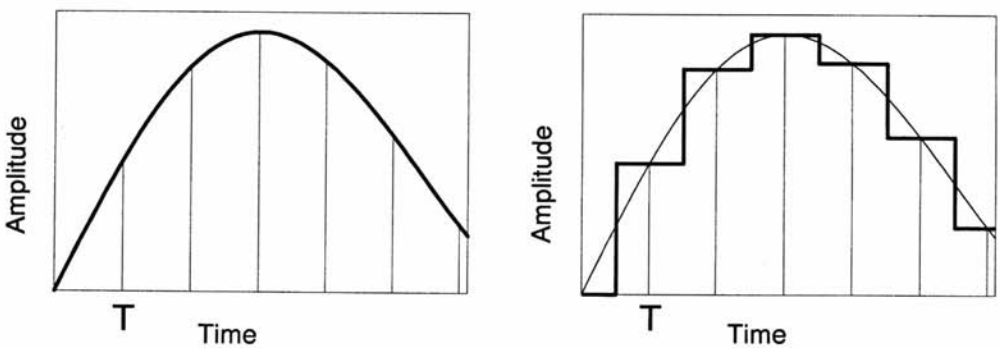


Figure 3.6. Sampling and reconstruction of a slowly varying signal. The continuously varying signal on the left is sampled at discrete time intervals T. The reconstructed signal is shown overlaid on the original in the right figure.

Ex.1018 / Page 65 of 343
AMAZON.COM, INC.

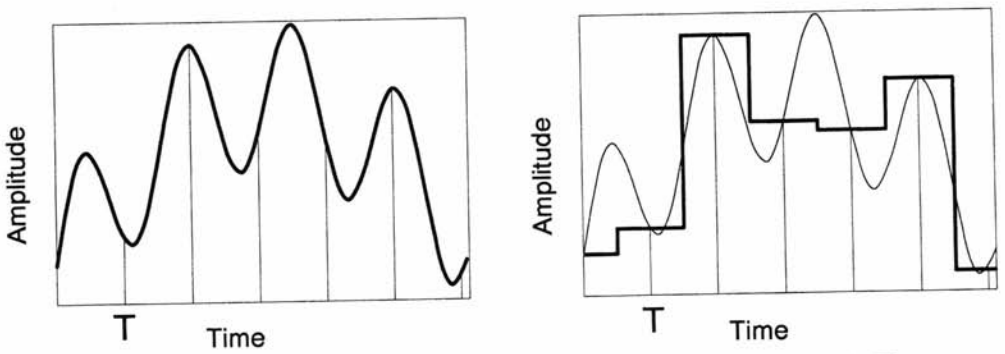


Figure 3.7. Inadequate reconstruction of a higher frequency signal. The original signal varies more rapidly than that in Figure 3.6, and the resulting reconstruction on the right shows gross errors.

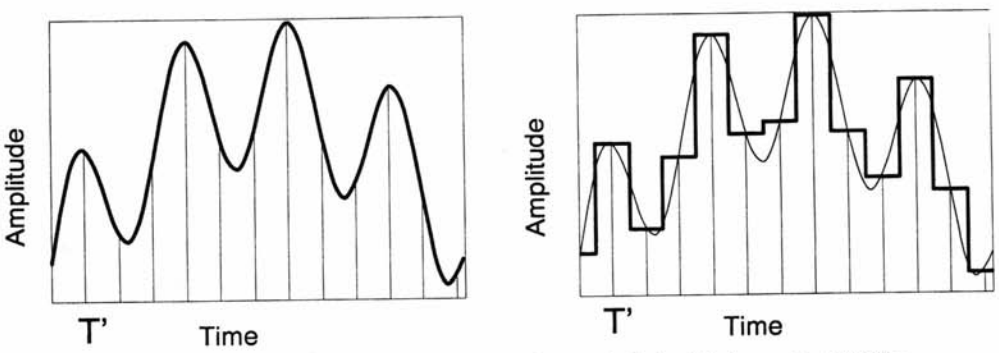


Figure 3.8. The higher frequency signal sampled at a higher rate, **T'**. The reconstructed signal now more accurately tracks the original.

signal are obtained. If the signal varies more rapidly (i.e., has components of a higher frequency), the same sampling rate may not correctly capture the signal (see Figure 3.7). Instead a higher sampling rate (smaller **T'**) is required to capture the more rapidly changing signal (see Figure 3.8); this signal has a higher **bandwidth.**

Bandwidth is the range of frequencies contained in a signal. For speech these span from a low frequency at the fundamental frequency of voicing (perhaps 100 to 400 Hertz—lower for men, higher for women, and higher still for children) to upwards of 15 kHz. The telephone network supports a voice bandwidth of approximately 3100 Hz (see Chapter 10 for more details); speech is sampled at 8000 Hz for digital telephone transmission.[4] Although this bandwidth is adequate for intelligibility, the transmitted voice is clearly of lower quality than face-to-face

[4]If ideal filters really existed, this sampling rate could be reduced to 6200Hz (twice 3100 Hz), but real filters have a gradual frequency cutoff as opposed to the discontinuous response of ideal filters.

Ex.1018 / Page 66 of 343
AMAZON.COM, INC.

communication. Compare this with a compact audio disc, which samples at 44.1 kHz giving a frequency response in the range of 20 kHz. Although the higher bandwidth is required more for the musical instruments than for the singers, the difference in voice quality offered by the increased bandwidth response is striking as well. Of course the compact disc requires a sampling rate five times greater than the telephone with the resulting increased requirements for storage and transmission rate.

Sampling rate is only one consideration of digitization; the other is the resolution of the samples at whatever rate they are taken. Each sample is represented as a digital value, and that number covers only a limited range of discrete values with reference to the original signal. For example, if 4 bits are allowed per sample, the sample may be one of 16 values; while if 8 bits are allowed per sample, it may be one of 256 values. The size in terms of number of bits of the samples can be thought of as specifying the fineness of the spacing between the lines in Figure 3.9. The closer the spacing between possible sample values, the higher the resolution of the samples and the more accurately the signal can be reproduced. In Figure 3.9, the **error,** or difference between the original and reconstructed signals, is shaded; higher resolution results in decreased quantization error.

The resolution of the quantizer (the number of bits per sample) is usually described in terms of a **signal-to-noise ratio,** or **SNR.** The error between the quantized sample and the original value of the signal can be thought of as noise, i.e., the presence of an unwanted difference signal that is not correlated with the original. The higher the SNR, the greater the fidelity of the digitized signal. SNR for PCM encoding is proportional to two to the power of the number of bits per sample, i.e., SNR $\approx 2^B$ where B is bits per sample. Adding one bit to the quantizer doubles the SNR.

Note that the SNR, or quantization error, is independent of the sampling rate limitation on bandwidth. The data rate of a sampled signal is the product of the sample size times the sampling rate. Increasing either or both increases

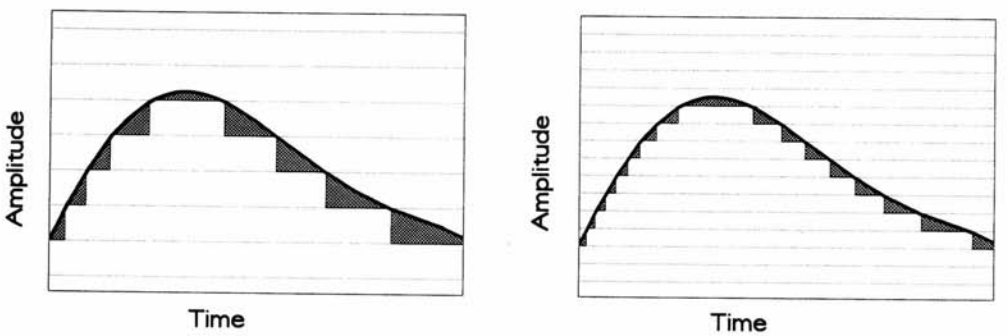


Figure 3.9. Low and high resolution quantization. Even when the signal is sampled continuously in time, its amplitude is quantized, resulting in an error, shown by the shaded regions.

the data rate. But increasing the sampling rate has no effect on quantization error and vice versa.

To compare the telephone and the compact audio disc again, consider that the telephone uses 8 bits per sample, while the compact disc uses 16 bits per sample. The telephone samples at 8 kHz (i.e., 8000 samples per second), while the compact disc sampling rate is 44.1 kHz. One second of telephone-quality speech requires 64,000 bits, while one second of compact disc-quality audio requires an order of magnitude more data, about 700,000 bits, for a single channel.[5]

Quantization and sampling are usually done with electronic components called **analog-to-digital** (A/D) and **digital-to-analog** (D/A) converters. Since one of each is required for bidirectional sampling and reconstruction, they are often combined in a **codec** (short for coder/decoder), which is implemented as a single integrated circuit. Codecs exist in all digital telephones and are being included as standard equipment in a growing number of computer workstations.

SPEECH CODING ALGORITHMS

The issues of sampling and quantization apply to any time-varying digital signal. As applied to audio, the signal can be any sound: speech, a cat's meow, a piano, or the surf crashing on a beach. If we know the signal is speech, it is possible to take advantage of characteristics specific to voice to build a more efficient representation of the signal. Knowledge about characteristics of speech can be used to make limiting assumptions for waveform coders or models of the vocal tract for source coders.

This section discusses various algorithms for **coders** and their corresponding **decoders.** The coder produces a digital representation of the analog signal for either storage or transmission. The decoder takes a stream of this digital data and converts it back to an analog signal. This relationship is shown in Figure 3.10. Note that for simple PCM, the coder is an A/D converter with a clock to generate a sampling frequency and the decoder is a D/A converter. The decoder can derive a clock from the continuous stream of samples generated by the coder or have its own clock; this is necessary if the data is to be played back later from storage. If the decoder has its own clock, it must operate at the same frequency as that of the coder or the signal is distorted.[6]

Waveform Coders

An important characteristic of the speech signal is its dynamic range, the difference between the loudest and most quiet speech sounds. Although speech has a

[5]The compact disc is stereo; each channel is encoded separately, doubling the data rate given above.

[6]In fact it is impossible to guarantee that two independent clocks will operate at precisely the same rate. This is known as clock skew. If audio data is arriving in real time over a packet network, from time to time the decoder must compensate by skipping or inserting data.

Ex.1018 / Page 68 of 343
AMAZON.COM, INC.

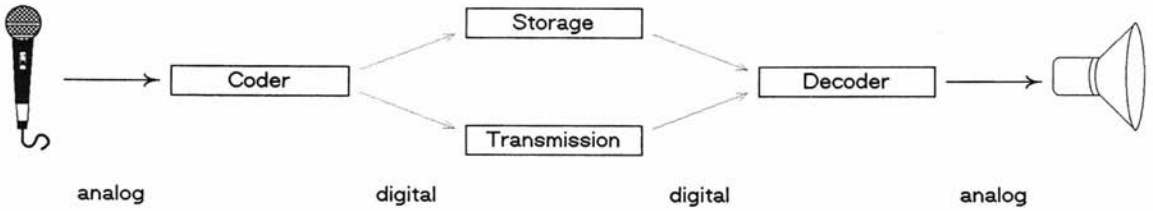


Figure 3.10. The coder and decoder are complementary in function.

fairly wide dynamic range, it reaches the peak of this range infrequently in normal speech. The amplitude of a speech signal is usually significantly lower than it is at occasional peaks, suggesting a quantizer utilizing a nonlinear mapping between quantized value (output) and sampled value (input). For very low signal values the difference between neighboring quantizer values is very small, giving fine quantization resolution; however, the difference between neighbors at higher signal values is larger, so the coarse quantization captures a large dynamic range without using more bits per sample. The mapping with nonlinear quantization is usually logarithmic. Figure 3.11 illustrates the increased dynamic range that is achieved by the logarithmic quantizer.

Nonlinear quantization maximizes the information content of the coding scheme by attempting to get roughly equal use of each quantizer value averaged over time. With linear coding, the high quantizer values are used less often; logarithmic coding compensates with a larger step size and hence coarser resolution at the high extreme. Notice how much more of the quantizer range is used for the lower amplitude portions of the signal shown in Figure 3.12. Similar effects can also be accomplished with adaptive quantizers, which change the step size over time.

A piece-wise linear approximation of logarithmic encoding, called **μ-law,**[7] at 8 bits per sample is standard for North American telephony.[8] In terms of signal-to-noise ratio, this 8 bits per sample logarithmic quantizer is roughly equivalent to 12 bits per sample of linear encoding. This gives approximately 72 dB of dynamic range instead of 48 dB, representing seven orders of magnitude of difference between the highest and lowest signal values that the coder can represent.

Another characteristic of speech is that adjacent samples are highly correlated; neighboring samples tend to be similar in value. While a speech signal contains frequency components up to one-half the sampling rate, the higher-frequency components are of lower energy than the lower-frequency components. Although the signal can change rapidly, the amplitude of the rapid changes is less than that of slower changes. This means that differences between adjacent samples are less than the overall range of values supported by the quantizer so fewer bits are required to represent differences than to represent the actual sample values.

[7]Pronounced "mu-law."

[8]A very similar scale, "A-law," is used in Europe.

Ex.1018 / Page 69 of 343
AMAZON.COM, INC.

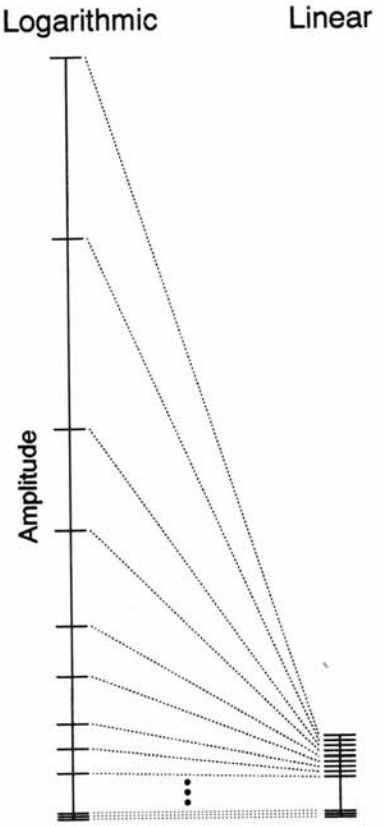


Figure 3.11. A comparison of dynamic range of logarithmic and linear coders for the same number of bits per sample. The logarithmic coder has a much greater dynamic range than a linear coder with comparable resolution at low amplitudes.

Encoding of the differences between samples is called **Delta PCM.** At each sampling interval the encoder produces the difference between the previous and the current sample value. The decoder simply adds this value to the previous value; the previous value is the only information it needs to retain. This process is illustrated in Figure 3.13.

The extreme implementation of difference coding is **Delta Modulation,** which uses just one bit per sample. When the signal is greater than the last sample, a "1" bit results; when less, a "0" bit results. During reconstruction, a constant step size is added to the current value of the signal for each "1" and subtracted for each "0."

This simple coder has two error modes. If the signal increases or decreases more rapidly than the step size times the sampling rate, **slope overload** results as the coder cannot keep up. This may be seen in the middle of Figure 3.14. To

Ex.1018 / Page 70 of 343
AMAZON.COM, INC.

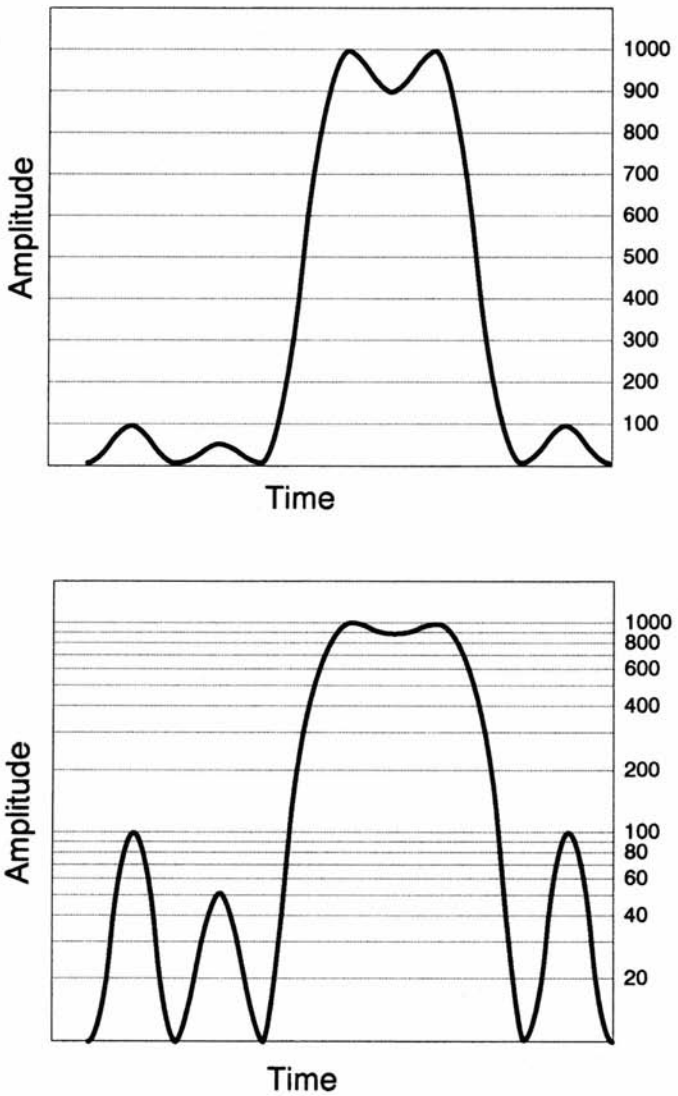


Figure 3.12. Use of a logarithmic scale disperses the signal density over the range of the coder. The top figure shows a linearly encoded signal; the bottom shows the same signal logarithmically encoded.

compensate for this effect, sampling rates significantly in excess of the Nyquist rate (e.g., in the range of 16 to 32 kHz) are usually used. The second error mode occurs when the delta-modulated signal has a low value such as in moments of silence. The reconstructed signal must still alternate up and down (see the right side of Figure 3.14) as the 1-bit coder cannot indicate a "no-change" condition. This is called **quieting noise.**

Ex.1018 / Page 71 of 343
AMAZON.COM, INC.

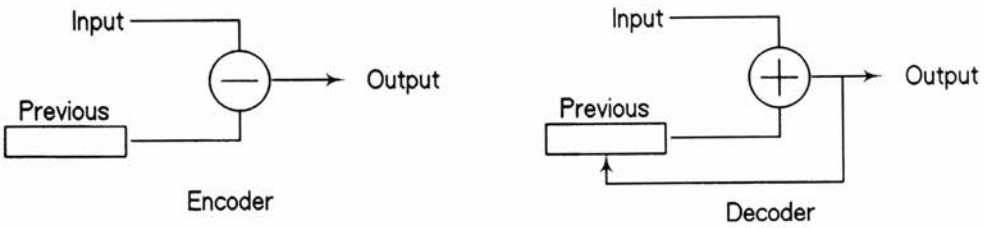


Figure 3.13. The Delta PCM encoder transmits the difference between sample values. The signal is reconstructed by the encoder, which always stores the previous output value.

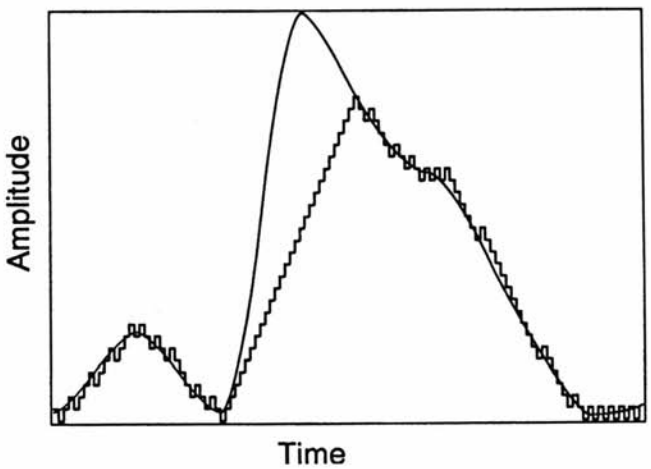


Figure 3.14. Delta Modulation. The jagged line is the delta modulated version of the smooth signal it encodes.

To compensate for these effects, **Continuously-Varying Slope Delta Modulation (CVSD)** uses a variable step size. Slope overload is detected as a series of consecutive "1" or "0" values. When this happens, the step size is increased to "catch up" with the input (see Figure 3.15). Note that the step size is increased at both the coder and the decoder, which remain synchronized because the decoder tracks the coder's state by its output; the sequence of identical bit values triggers a step size change in both.

After the step size increases, it then decays over a short period of time. Both the encoder and decoder must use the same time constant for this decay; they remain in synchronization because the decoder uses the incoming bit stream as its clock. Quieting noise is reduced by detecting the quieting pattern of alternating "0" and "1" values and decreasing the step size to an even smaller value while this condition persists.

CVSD is an example of *adaptive* quantization, so called because the quantizer adapts to the characteristics of the input signal. During periods of low signal

Ex.1018 / Page 72 of 343
AMAZON.COM, INC.

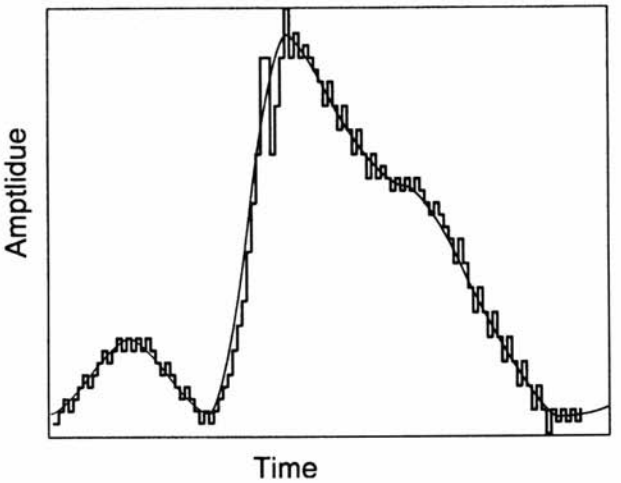


Figure 3.15. Quantizer step size increases for CVSD encoding to catch up with a rapidly increasing signal. The CVSD encoded version more closely tracks the original signal than does the delta modulation shown in Figure 3.14.

energy, a fine step size quantizer with a small range is employed, while a larger step size is substituted during higher amplitude signal excursions. In other words, at times when the signal is changing rapidly, there will be a larger difference between samples than at times when the signal is not changing rapidly. With CVSD these conditions are detected by monitoring the pattern of output bits from the quantizer.

A similar but more powerful encoder is **Adaptive Delta PCM (ADPCM),** a popular encoding scheme in the 16,000 to 32,000 bits per second range. ADPCM is a Delta PCM encoder in that it encodes the difference between samples but does so using a method in which the same number of bits per sample can sometimes represent very small changes in the input signal and at other times represent much larger changes. The encoder accomplishes this by looking up the actual sample differences in a table and storing or transmitting just enough information to allow the decoder to know where to look up a difference value from the same table.

When it samples the current input value of the signal, an ADPCM coder uses two additional pieces of internal information in addition to the table to decide how to encode the signal. This information is a function of the history of the signal, and therefore is available to the decoder even though it is not transmitted or stored. The first stored value is the *state,* which is just a copy of the last value output from the coder; this is common to all difference encoders. The second value is the *predictor,* perhaps more appropriately called an adapter, which directly compensates for changes in the signal magnitude.

The key component of the ADPCM decoder is its output table. Although the difference signal is encoded in only four bits (including a sign bit to indicate an increasing or decreasing output value), the table is much "wider." Each entry in

Ex.1018 / Page 73 of 343
AMAZON.COM, INC.

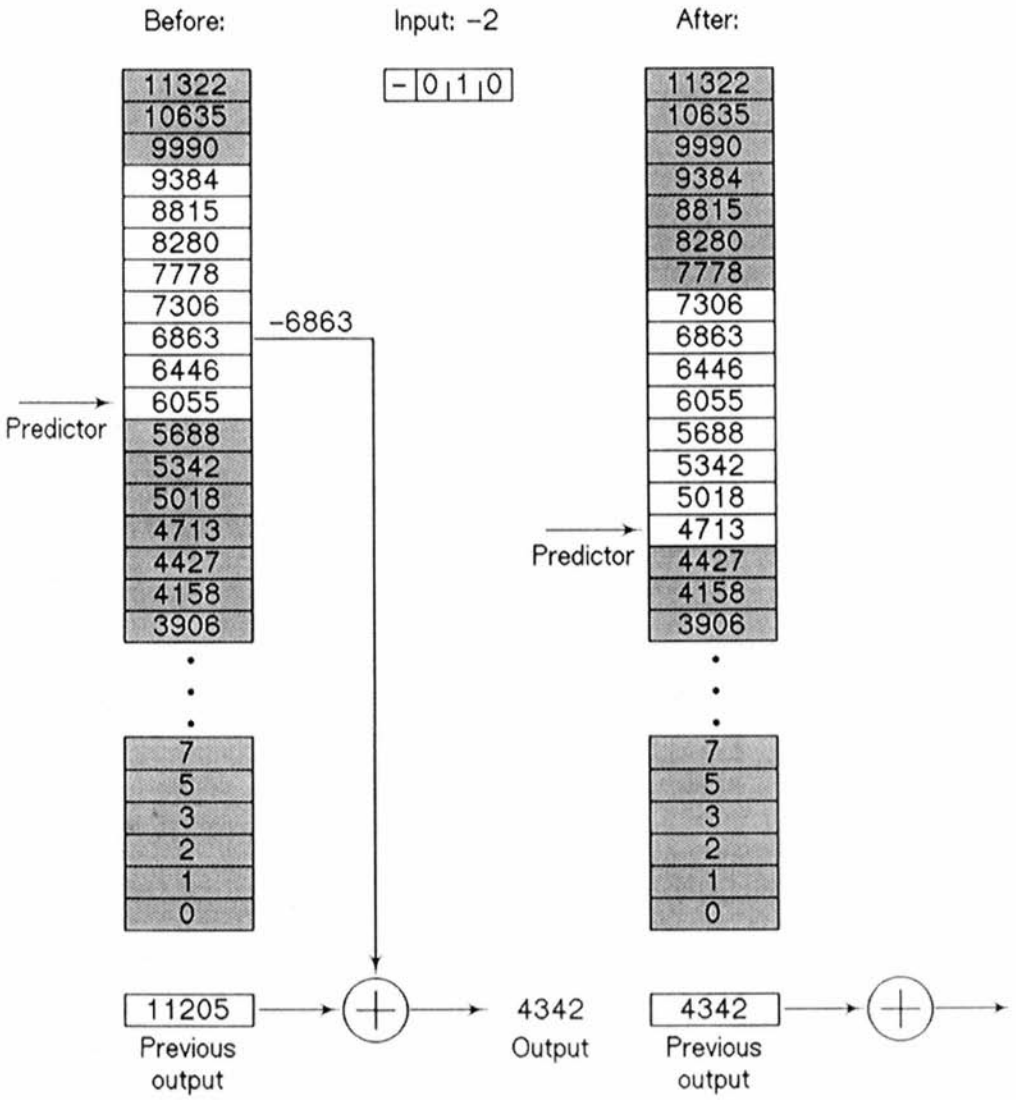


Figure 3.16. An ADPCM decoder.

the table may be eight or twelve bits giving a much larger dynamic range. At any moment, the three-bit difference value can be used to look up one of eight values from the table. These values are then added to or subtracted from the last output value giving the new output value. But the table has many more than eight entries. The predictor indicates where in the table the range of eight possible difference values should be chosen. As a function of the signal, the predictor can move to a coarser or finer part of the table (which is approximately exponential) as the signal amplitude changes. This movement within the lookup table is how adaptation occurs.

Ex.1018 / Page 74 of 343
AMAZON.COM, INC.

An ADPCM decoder is shown in Figure 3.16. For each input sample, the ADPCM decoder looks up a difference value that it adds to (or subtracts from) the previous output value to generate the new output value. It then adapts its own internal state. The difference value is found by table lookup relative to the predictor, which is just a base pointer into the table. This base pointer specifies the range of values which the encoder output can reference. Depending on the sign bit of the encoder output, the value found at the specified offset from the predictor is then added to or subtracted from the previous output value.

Based on another table, the input number is also used to look up an amount to move the base pointer. Moving the base pointer up selects a region of the lookup table with larger differences and coarser spacing; moving down has the opposite effect. This is shown on the right side of Figure 3.16 and is the adaptation mechanism for ADPCM.

Using an ADPCM encoding scheme with 4 bit samples at an 8 kHz sampling rate, it is possible to achieve speech quality virtually indistinguishable from that of 64,000 bit per second log PCM using only half of that data rate. Since many telephone channels are configured for 64,000 bits per second, ADPCM is used as part of a new approach to higher quality telephony that samples twice as often (16,000 times per second), thereby using the full digital telephone channel bandwidth to give a speech bandwidth of approximately 7.1 kHz. (The full algorithm is described in [Mermelstein 1975].) This has become an international standard because of its usefulness in telephone conference calls; the increased bandwidth is particularly helpful in conveying cues as to the identity of the speaker.[9]

In summary, waveform coders provide techniques to encode an input waveform such that the output waveform accurately matches the input. The simplest waveform coders are suitable for general audio encoding, although music sources place greater requirements on the encoder than speech. Due to the larger dynamic range of most music and its greater bandwidth, adequately digitizing music requires a larger sample size and a higher sampling rate. However, more sophisticated waveform coders take advantage of statistical properties of the speech signal to encode the signal in fewer bits without a loss of intelligibility. Because these methods depend on characteristics specific to speech, they are often not adequate for digitizing nonspeech audio signals.

Source Coders

In contrast to the waveform coders, that compress the speech signal by encoding the waveform, the source coders model the acoustical conditions that produced the signal. Source coders assume the sound was produced by the vocal tract and analyze it according to a source model. This model consists of an excitation component and a filter component corresponding to the vocal tract. Source decoders

[9]CCITT specification G.722. The **CCITT** is the International Telephone and Telegraph Consultative Committee, which defines international telecommunications standards.

Ex.1018 / Page 75 of 343
AMAZON.COM, INC.

then reconstruct the sound not by reproducing the waveform of the signal but rather by generating a different signal that has the same spectral characteristics as the original.

The most common source coder is **Linear Predictive Coding (LPC).** LPC analysis requires somewhat expensive processing for the encoder but the decoder is fairly simple. An input signal is analyzed to determine whether it is periodic (whether the speech is voiced or not) and, if so, its pitch is computed. The coder then examines the spectrum of the input signal and determines a fixed number of resonances as parameters of a filter bank. The filter is an "all pole" model, which means it can reproduce only the peaks of the spectrum of the vocal tract. These parameters are predictor coefficients, which indicate how a sample at any moment in time depends on the previous samples. An overall gain value is also computed from the average magnitude or energy of the input. The coder updates this set of values at a regular rate, typically every 20 milliseconds.

The decoder (see Figure 3.17) models the vocal tract to produce a steady stream of digital waveform values. The model consists of a sound source and a filter and is controlled by the parameters derived by the encoder including pitch, voiced/unvoiced switch, gain, and filter coefficients. The source may be periodic with a range of possible pitches or it may be aperiodic. To output a waveform value, a new value is read from the input and scaled by a gain factor; then it undergoes a filtering process that can be represented by the weighted sum of the current sample and the previous N samples, where N is the number of predictor coefficients. Note that this system left alone would produce a continuous stream of steady output; the parameters determine what the output sounds like. But the model is updated with each new set of vocal tract parameters frequently enough to make it sound like speech.

The LPC decoder requires significantly less computation than the encoder, which must perform sophisticated analysis of the speech signal. This assymetry may be useful in mass-market consumer applications, which store voice prompts

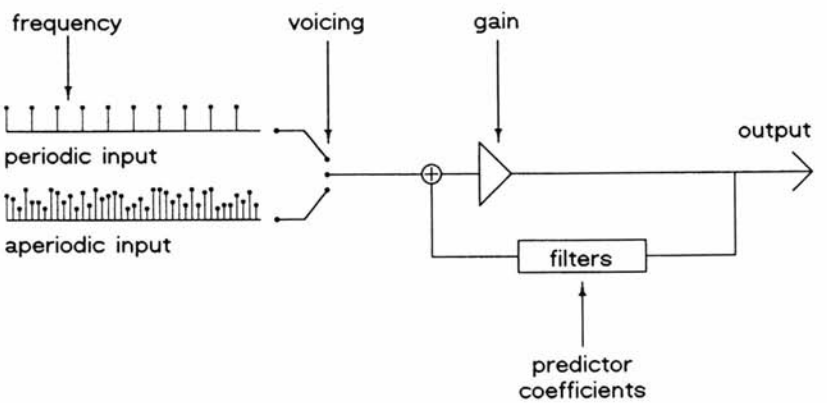


Figure 3.17. The LPC decoder models the vocal tract.

Ex.1018 / Page 76 of 343
AMAZON.COM, INC.

in read only memory (ROM). If the product need not encode speech but just play it, an expensive encoder is unnecessary. Such assymetry is not an advantage for bidirectional applications such as telecommunications. Digitization algorithms that employ such different encoders and decoders are sometimes called **analysis-synthesis techniques,** which are often confused with text-to-speech synthesis; this is the topic of Chapter 5.

Speech can be encoded intelligibly using LPC at around 2400 bits per second. This speech is not of high enough quality to be useful in many applications and with the decreasing cost of storage has lost much of its original attractiveness as a coding medium. However, since LPC compactly represents the most salient acoustical characteristics of the vocal tract, it is sometimes used as an internal representation of speech for recognition purposes. In addition, a related source coding algorithm may emerge as the standard for the digital cellular telephone system.

A number of techniques have been developed to increase the quality of LPC encoding with higher data rates. In addition to the LPC signal, **Residually Excited LPC (RELP)** encodes a second data stream corresponding to the difference that would result between the original signal and the LPC synthesized version of the signal. This "residue" is used as the input to the vocal tract model by the decoder, resulting in improved signal reproduction.

Another technique is **Multipulse LPC,** which improves the vocal tract model for voiced sounds. LPC models this as a series of impulses, which are very short-duration spikes of energy. However, the glottal pulse is actually triangular in shape and the shape of the pulse determines its spectrum; Multipulse LPC approximates the glottal pulse as a number of impulses configured so as to produce a more faithful source spectrum.

The source coders do not attempt to reproduce the waveform of the input signal; instead, they produce another waveform that sounds like the original by virtue of having a similar spectrum. They employ a technique that models the original waveform in terms of how it might be produced. As a consequence, the source coders are not very suitable for encoding nonspeech sounds. They are also rather fragile, breaking down if the input includes background noise or two people talking at once.

CODER CONSIDERATIONS

The primary consideration when comparing the various speech coding schemes is the tradeoff between information rate and signal quality. When memory and/or communication bandwidth are available and quality is important, as in voice mail systems, higher data-rate coding algorithms are clearly desirable. In consumer products, especially toys, lower quality and lower cost coders are used as the decoder and speech storage must be inexpensive. This section discusses a number of considerations for matching a coding algorithm to a particular application environment.

Ex.1018 / Page 77 of 343
AMAZON.COM, INC.

Intelligibility

The primary consideration in choosing a coding scheme is that of intelligibility—*everyone* wants the highest quality speech possible. Some coding algorithms manage to reduce the data rate at little cost in intelligibility by using techniques to get the most out of each bit. In general, however, there is a strong tradeoff between speech quality and data rate. For any coding scheme there are likely to be options to use less bits for each value or to sample the audio less frequently at the cost of decreased intelligibility.

There are actually two concepts to consider: signal **error** and **intelligibility.** Error is a mathematically rigorously defined measure of the energy of the difference between two signals on a sample-by-sample basis. Error can be measured against waveforms for the waveform coders and against spectra for the source coders. But this error measure is less important than intelligibility for many applications. To measure intelligibility, human subjects must listen to reproduced speech and answer questions designed to determine how well the speech was understood. This is much more difficult than comparing two signals. The discussion of the intelligibility of synthetic speech in Chapter 9 describes some listening tests to measure intelligibility.

A still more evasive factor to measure is signal **quality.** The speech may be intelligible but is it pleasant to hear? Judgment on this is subjective but can be extremely important to the acceptance of a new product such as voice mail, where use is optional for the intended customers.

A related issue is the degree to which any speech coding method interferes with listeners' ability to identify the talker. For example, the limited bandwidth of the telephone sometimes makes it difficult for us to quickly recognize a caller even though the bandwidth is adequate for transmitting highly intelligible speech. For some voice applications, such as the use of a computer network to share voice and data during a multisite meeting, knowing who is speaking is crucial.

Editing

The ease with which a speech signal can be edited by cutting and splicing the sequence of stored data values may play a role in the choice of an encoding scheme. Both delta (difference) and adaptive coders track the current state during signal reproduction; this state is a function of a series of prior states, i.e., the history of the signal, and hence is computed each time the sound is played back. Simply splicing in data from another part of a file of speech stored with these coders does not work as the internal state of the decoder is incorrect.

Consider an ADPCM-encoded sound file as an example. In a low energy (quiet) portion of the sound, the predictor (the base pointer into the lookup table) is low so the difference between samples is small. In a louder portion of the signal the predictor points to a region of the table with large differences among samples. However, the position of the predictor is not encoded in the data stream. If samples from the loud portion of sound are spliced onto the quiet portion, the predictor will be incorrect and the output will be muffled or otherwise distorted

Ex.1018 / Page 78 of 343
AMAZON.COM, INC.

as the wrong differences will be accessed from the lookup table to produce the coder output.

With coders such as ADPCM predictable results can be achieved if cutting and splicing is forced to occur during silent parts of the signal; during silent portions the predictor returns to a known state, i.e., its minimum value. As long as enough low value samples from the silent period are retained in the data stream so as to force the predictor down to its minimum, another speech segment starting from the minimum value prediction (just after silence) can be inserted. This precludes fine editing at the syllable level but may suffice for applications involving editing phrases or sentences.

Adaptation is not an issue with PCM encoding as the value of each sample is independent of any prior state. However, when pieces of PCM data are spliced together, the continuity of the waveform is momentarily disrupted, producing a "click" or "pop" during playback. This is due to changing the *shape* of the waveform, not just its value, from sample to sample. The discontinuity is less severe when the splice takes place during low energy portions of the signal, since the two segments are less likely to represent drastically different waveform shapes. The noise can be further reduced by both interpolating across the splice and reducing the amplitude of the samples around the splice for a short duration.

A packetization approach may make it possible to implement cut-and-paste editing using coders that have internal state. For example, LPC speech updates the entire set of decoder parameters every 20 milliseconds. LPC is really a stream of data packets, each defining an unrelated portion of sound; this allows edits at a granularity of 20 milliseconds. The decoder is designed to interpolate among the disparate sets of parameter values it receives with each packet allowing easy substitution of other packets while editing.

A similar packetization scheme can be implemented with many of the coding schemes by periodically including a copy of the coder's internal state information in the data stream. With ADPCM for example, the predictor and output signal value would be included perhaps every 100 milliseconds. Each packet of data would include a header containing the decoder state, and the state would be reset from this data each time a packet was read. This would allow editing to be performed on chunks of sound with lengths equal to multiples of the packet size.

In the absence of such a packetization scheme, it is necessary to first convert the entire sound file to PCM for editing. Editing is performed on the PCM data and then is converted back to the desired format by recoding the signal based on the new data stream.

Silence Removal

In spontaneous speech there are substantial periods of silence.[10] The speaker often pauses briefly to gather his or her thoughts either in midsentence or

[10]Spontaneous speech is normal unrehearsed speech. Reading aloud or reciting a passage from memory are not spontaneous.

Ex.1018 / Page 79 of 343
AMAZON.COM, INC.

between sentences. Although the term "sentence" may be used fairly loosely with spontaneous speech, there are speech units marked by pauses for breath that usually correspond roughly to clauses or sentences. Removing these silences during recording can save storage space. Such techniques are commonly applied to almost any encoding technique. For example, silence removal is used with most voice mail systems regardless of the encoding scheme. The amount of storage space saved by silence removal is directly proportional to the amount of speech that was pauses.

A form of silence removal is actually used in many telephone networks. There is no need to keep a transmission channel occupied transferring the audio data associated with the frequent pauses in a conversation. By detecting these pauses, the channel may be freed to carry another conversation for the duration of the pause. When one of the parties starts speaking again, the channel must be restored to transmit the conversation. In practice, a small number of channels are used to carry a larger number of conversations; whenever a new spurt of speech occurs during one of the conversations, a free channel is assigned to it.

This channel-sharing scheme was originally implemented as an analog system known as **Time Assigned Speech Interpolation (TASI)** [O'Neill 1959] to carry an increased number of calls over limited-capacity, trans-Atlantic telephone cables. Today, digital techniques are employed to implement "virtual circuits," so called because the actual channel being used to carry a specific conversation will vary over time. The channel refers to a particular slot number in a digital data stream carrying a large number of conversations simultaneously. During silence, the slot associated with the conversation is freed and made available to the next conversation that begins. When the first conversation resumes, the slot it formerly occupied may be in use, in which case it is assigned to a new slot. These adjustments are transparent to equipment outside of the network. The number of channels required to carry the larger number of conversations is a property of the statistical nature of the speaking and silence intervals in ordinary conversations and is an important aspect of network traffic management.

During a telephone conversation, no time compression is achieved by silence removal; although audio data may not be transmitted across the network during silences, the listener must still wait for the talker to resume. In a recorded message with pauses removed, less time is required for playback than transpired during recording. Silence removal is generally effective, but some of the pauses in speech contain meaning; part of the richness of the message may be the indication of the talker's thought process and degree of deliberation implied by the location and duration of the pauses.

Several techniques can be used to reconstruct the missing pauses. Instead of simply removing the pauses from the stored data, a special data value may be inserted into the file indicating the presence of a pause and its duration. During playback this information indicates to the decoder to insert the right amount of silence. Since the ambient (background) noise of the recording is replaced by absolute silence, an audible discontinuity is noticeable. Another technique involves recording a small amount of the background noise and then playing it

Ex.1018 / Page 80 of 343
AMAZON.COM, INC.

over and over to fill the reconstructed pauses. It is necessary to capture enough of the silence that the listener cannot easily hear the repetitive pattern during playback.

Another difficulty with silence removal is proper detection of the beginning, or onset, of speech after silence. The beginnings and endings of speech segments are often of low amplitude, which may make them difficult to detect, yet they contain significant phonetic information. Endings can be easily preserved simply by recording a little extra "silence" at the end of each burst of speech. Beginnings are harder; the encoder must buffer a small amount of the signal (several hundred milliseconds) so that when speech is finally detected this recently-saved portion can be stored with the rest of the signal. This buffering is difficult enough that it is often not done, degrading the speech signal.

Time Scaling

Because speech is slow from the perspective of the listener, it is desirable to be able to play it back in less time than required to speak it originally. In some cases one may wish to slow down playback, perhaps while jotting down a phone number in a rapidly-spoken message. Changing the playback rate is referred to as **time compression** or **time-scale modification,** and a variety of techniques of varying complexity are available to perform it.[11]

Skipping pauses in the recording provides a small decrease in playback time. The previous section discussed pause removal from the perspective of minimizing storage or transmission channel requirements; cutting pauses saves time during playback as well. It is beneficial to be selective about how pauses are removed. Short (less than about 400 millisecond) **hesitation pauses** are usually not under the control of the talker and can be eliminated. But longer **juncture pauses** (about 500 to 1000 milliseconds) convey syntactic and semantic information as to the linguistic content of speech. These pauses, which often occur between sentences or clauses, are important to comprehension and eliminating them reduces intelligibility. One practical scheme is to eliminate hesitation pauses during playback and shorten all juncture pauses to a minimum amount of silence of perhaps 400 milliseconds.

The upper bound on the benefit to be gained by dropping pauses is the total duration of pauses in a recording, and for most speech this is significantly less than 20% of the total. Playback of nonsilent portions can be sped up by increasing the clock rate at which samples are played, but this produces unacceptable distortion. Increasing the clock shifts the frequency of speech upwards and it becomes whiny, the "cartoon chipmunk" effect.

A better technique, which can be surprisingly effective if implemented properly, is to selectively discard portions of the speech data during playback. For example,

[11]For a survey of time-scale modification techniques, see [Arons 1992a].

Ex.1018 / Page 81 of 343
AMAZON.COM, INC.

for an increase in playback rate of 20%, one could skip the first 12 milliseconds of every 60 milliseconds of data. This was first discussed by Maxemchuk [Maxemchuk 1980] who refined it somewhat by attempting to drop lower-energy portions of the speech signal in preference to higher energy sections. This technique is rather sensitive to the size of the interval over which a portion of the sound is dropped (60 milliseconds in the above example); this interval should ideally be larger than a pitch period but shorter than a phoneme.

Time compression by this method exhibits a burbling sound but can be surprisingly intelligible. The distortion can be somewhat reduced by smoothing the junctions of processed segments, by dropping and then increasing amplitude rapidly across the segment boundaries, or by performing a cross-fade in which the first segment decays in amplitude as the second segment starts up.

A preferred method of time compression is to identify pitch periods in the speech signal and remove whole periods. To the extent that the signal is periodic, removing one or more pitch periods should result in an almost unchanged but time-compressed output. The most popular of these time domain algorithms is **Synchronous Overlap and Add (SOLA)** [Roucos and Wilgus 1985]. SOLA removes pitch periods by operating on small windows of the signal, shifting the signal with respect to itself and finding the offset with the highest correlation, and then averaging the shifted segments. Essentially, this is pitch estimation by autocorrelation (finding the closest match time shift) and then removal of one or more pitch periods with interpolation. SOLA is well within the capacity of many current desktop workstations without any additional signal processing hardware.

Time compression techniques such as SOLA reduce the intelligibility of speech, but even naive listeners can understand speech played back twice as fast as the original; this ability increases with listener exposure of 8 to 10 hours [Orr *et al.* 1965]. A less formal study found listeners to be uncomfortable listening to normal speech after one-half hour of exposure to compressed speech [Beasley and Maki 1976].

Robustness

A final consideration for some applications is the robustness of the speech encoding scheme. Depending on the storage and transmission techniques used, some data errors may occur, resulting in missing or corrupted data. The more highly compressed the signal, the more information is represented by each bit and the more destructive any data loss may be.

Without packetization of the audio data and saving state variables in packet headers, the temporary interruption of the data stream for most of the adaptive encoders can be destructive for the same reasons mentioned in the discussion of sound editing. Since the decoder determines the state of the encoder simply from the values of the data stream, the encoder and decoder will fall out of synchronization during temporary link loss. If the state information is transmitted periodically, then the decoder can recover relatively quickly.

Another aspect of robustness is the ability of the encoder to cope with noise on the input audio signal. Since the encoders rely on properties of the speech signal

Ex.1018 / Page 82 of 343
AMAZON.COM, INC.

to reduce their bit rates, inputs that fail to obey these constraints will not be properly encoded. One should not expect to encode music with most of these encoders, for example. LPC is an extreme case as it is a source coder and the decoder incorporates a model of the vocal tract.

SUMMARY

This chapter has discussed digitization and encoding of speech signals. Digitization is the process whereby naturally analog speech is sampled and quantized into a digital form amenable to manipulation by a computer. This may be likened to the previous chapter's description of how the ear transduces variations in air pressure on the ear drum to electrical pulses on neurons connected to the brain. Digitization stores a sound for future rote playback, but higher level processing is required to understand speech by recognizing its constituent words.

This chapter has described several current speech coding schemes in some detail to illustrate how knowledge of the properties of speech can be used to compress a signal. This is not meant to be an exhaustive list, and new coding algorithms have been gaining acceptance even as this book was being written. The coders described here do illustrate some of the more common coders that the reader is likely to encounter. The design considerations across coders reveal some of the underlying tradeoffs in speech coding. Basic issues of sampling rate, quantization error, and how the coders attempt to cope with these to achieve intelligibility should now be somewhat clear.

This chapter has also mentioned some of the considerations in addition to data rate that should influence the choice of coder. Issues such as coder complexity, speech intelligibility, ease of editing, and coder robustness may come into play with varying importance as a function of a particular application. Many of the encoding schemes mentioned take advantage of redundancies in the speech signal to reduce data rates. As a result, they may not be at all suitable for encoding nonspeech sounds, either music or noise.

Digital encoding of speech may be useful for several classes of applications; these will be discussed in the next chapter. Finally, regardless of the encoding scheme, there are a number of human factors issues to be faced in the use of speech output. This discussion is deferred to Chapter 6 after the discussion of text-to-speech synthesis, as both recorded and synthetic speech have similar advantages and liabilities.

FURTHER READING

Rabiner and Schafer is the definitive text on digital coding of speech. Flanagan *et al.* [1979] is a concise survey article covering many of the same issues in summary form. Both these sources assume familiarity with basic digital signal processing concepts and mathematic notation.

Ex.1018 / Page 83 of 343
AMAZON.COM, INC.

4

Applications and Editing of Stored Voice

The previous chapter described some of the basic techniques for digitizing and encoding speech for either transmission or storage but said little of its role as a computer data type. This chapter focuses on computer applications of stored speech with particular emphasis on the underlying system support they require, especially sound storage facilities and editing. First, a taxonomy of voice applications is described based on the usage characteristics of situations in which speech is digitized and played back. Next, one of these classes—speech in documents—is explored in some detail. Finally, this chapter presents underlying representations and a number of examples of speech editors, which are required to support sophisticated use of audio in documents. In the course of this discussion, many examples of research projects utilizing digitized speech are mentioned to illustrate the range of approaches that have been taken to applications of stored voice.

This chapter focuses on application areas, their associated system requirements, and the functionality they require for supporting voice digitization and playback. Discussion of many user interaction requirements and techniques is postponed until Chapter 6, which deals with user interface issues for interactive speech response systems that employ both digitized and synthesized speech. For the purposes of this chapter, it is assumed that each application runs in isolation; Chapter 12 discusses issues of voice as a data type, including visual representations and sharing of voice data among applications on a computer workstation.

Ex.1018 / Page 84 of 343
AMAZON.COM, INC.

TAXONOMY OF VOICE OUTPUT APPLICATIONS

Stored voice is essential to a variety of applications such as these.

- Toys that play a small number of messages from read only memory.
- Dictation, in which voice is recorded for later transcription into text by another person.
- Voice mail, in which a message is recorded informally by a caller and is accessed once by the recipient over a telephone-based interface.
- Voice annotation of text or multimedia documents, which involve more careful authoring and often a visual user interface.

This section characterizes such applications according to the functionality that stored voice requires as differentiated by how users employ it in these classes of applications. Applications may emphasize either recording or playback, and the stored voice may be heard once or repeatedly. How the stored voice is to be used in a particular application determines the underlying system support required for both recording and playback as well as influences the choice of a particular speech coding technology and how it is stored as computer files.

Playback-Only Applications

Stored voice can be employed in a variety of applications for which a small repertoire of utterances needs to be spoken. Depending on the application, varying degrees of speech quality may be acceptable. Applications at the low-end of voice output technology employ single integrated circuits for both the storage and decoding of the digitized speech. Examples of this simple technology are inexpensive consumer products such as talking toys (dolls, stuffed animals, etc.) and video games. Some answering machines use stored speech to record the time onto the answering machine tape at the end of each incoming message. Answering machines that store the caller's message digitally are also appearing as consumer products. Voice warnings are used in both commercial and military aircraft cockpits. Automobile manufacturers have used digitized speech for warnings, and some cash registers recite each item and its price; however, neither of these two applications has proven to be very popular.

In situations where only minimal quality speech output is needed, it may be economical to use an asymmetric combination of speech coder and decoder. It is practical to use a computationally intensive coding algorithm, such as LPC, if the decoder is simple and inexpensive. Only the decoder needs to be included in a product which supports playback only.

Higher quality speech is needed for recorded public announcements such as the recitation of station names on automated trains or periodic, repetitious announcements over airport public address systems (e.g., warnings about taking baggage carts down the escalator). Because it is important for the listener to understand the message, these applications demand higher quality speech. In

Ex.1018 / Page 85 of 343
AMAZON.COM, INC.

Japan such announcements are ubiquitous in public locations and are usually spoken in a stylized female voice.

A similar class of applications are the so-called **audio-tex,** or **interactive voice response (IVR)** telephone-based information services. To use such services, one dials a telephone number and selects topics with touch tones. Recorded speech is used both for the information to be played back after selection as well as for prompts and menus. Examples include services to hear a weather forecast for a particular city, to find out one's current bank account balance, and to access an airline's flight schedule. The stored voice for these applications needs to be of adequate quality to be easily understood over the telephone by infrequent users. Because the service is accessed over the telephone, the sound data is most conveniently stored digitally on magnetic disk at a single site. Interaction techniques for interactive voice response systems will be discussed in detail in Chapter 6.

For all the applications just mentioned, the stored speech is used as "read only" data as the purpose of the application is to play a fixed set of prerecorded sounds. There is no user interaction with the voice itself; the system simply plays it at the appropriate time in response to the user's input. There is no need for a speech editing capability or a visual representation of the recording on a display, and the speech may be stored in whatever format is most convenient for the output device.

The developers of such applications occasionally need to update the voice recordings, and may use an editor or speech development system to do so. This development system does not need to be distributed with the application since updating can be done at a single, central location on an infrequent basis and distributed in ROM (Read Only Memory), floppy disks, or digital tape. Even for the developers, however, "editing" often consists of simply rerecording the passage repeatedly until the desired diction is achieved.

Some applications make use of a limited repertoire of "canned" speech segments to compose sentences; this involves concatenating short segments of digitized sound. One example is the mechanized playback of telephone numbers from directory assistance. It is essential that the individual sounds be accessible quickly enough during playback to avoid a pause between segments. If the speech is stored in semiconductor memory, this is not an issue, but for slower media, access time may be problematic.

In most of these applications, voice is chosen because it is the best output medium available. Because voice broadcasts through the air, it can be used in situations where displays would not be appropriate or visible. It is chosen as a warning channel because it is effective when the user is doing something else with his eyes, such as watching where he is driving or flying. In the case of the telephone-based services, access is also a dominant issue; users of such services take advantage of the nearly universal availability of the telephone. These advantages will be discussed in more detail in Chapter 6.

Interactive Record and Playback Applications

Voice mail is currently the most pervasive example of the class of application that involves frequent recording and playback of speech segments. Each time a call is

Ex.1018 / Page 86 of 343
AMAZON.COM, INC.

received, a new message is recorded. Usually the recipient listens to the message only once but may replay parts of the message while transcribing information such as the caller's telephone number. Although voice mail systems are most often treated simply as elaborate answering machines (with which messages are recorded once and played once), they are functionally more powerful than the consumer products found in homes. For example, voice mail systems can be used within an organization to disperse voice memos to distribution lists specifying multiple recipients or to specify a future delivery date for a message. This verbal memo is recorded just like a telephone message, although the author may speak more carefully or rerecord the message several times because it will be heard by a wider audience.

Voice is slow and serial, and telephone-based voice mail interfaces do not offer the user the opportunity to save messages in a "folder" with a mnemonic name as do most electronic mail systems. Archiving old voice messages and recalling them later is tedious, and few users take advantage of such capability; consequently, recording and playback occur equally often, which eliminates the advantages of an asymmetrical encoder/decoder combination. Random access of the recorded speech file is necessary as well to support repeated listening to a portion of the message.

Because of the more casual nature of telephone-based voice messages over paper memoranda, editing capabilities need not be provided; it would be difficult to build an effective editor using an audio-only interface based on touch tones. Instead, the user may be offered the option of reviewing a message before sending it and perhaps may choose to rerecord it. This is in contrast to the system recordings provided by the voice mail vendor, which are recorded very carefully often, with the assistance of voice editing software.

During recording, **Automatic Gain Control (AGC)** may be employed in an attempt to record constant amplitude messages from variable quality telephone connections. AGC uses variable amplification during recording to normalize the level of the recorded signal independent of the source level but it has limitations. Boosting the gain for a quiet speaker in a noisy environment may not enhance intelligibility as both the signal and noise are amplified; a better strategy is to monitor recording levels and prompt the quiet talker to speak louder. Another problem with AGC is that it tends to increase the gain during quiet portions of the message, or pauses, which causes a "whoosh" as the background noise increases; this is evident on many home videotapes. AGC can be improved by normalizing the gain based on peaks in the speech signal averaging over a time period, but this technique is not in widespread use. Although each telephone call can have a different audio level, the level is usually consistent for the duration of a message. This allows gain to be applied after the recording by multiplying each sample by the same scale factor; this avoids the whoosh produced when the gain is varied during a recording.

Dictation

Dictation is an application of stored voice in which recording capability is required by the user, but the stored voice is never meant to be heard by the ulti-

Ex.1018 / Page 87 of 343
AMAZON.COM, INC.

mate recipient. Instead, a transcriptionist listens to the recorded voice and transcribes it into text, often as a letter, memo, or short report.

Dictation has a different set of interaction requirements than other voice applications. The originator of a dictation spends a significant amount of time thinking about content and so requires a start/stop control or push-to-talk button; this is somewhat in contrast to telephone messages, where the caller must speak all at once at the "beep." The transcriptionist uses a foot pedal to control playback. Transcribing requires the capability to repeatedly review a portion of the message to understand individual words, but minimal random access to jump or scan through large portions of the recorded voice. The ultimate recipient of the message reads the transcription, having no interaction with the stored voice.

Although dictation and transcription have traditionally been done using analog tape recorders, other methods are now available. Dictation in an office environment can be digitized with the transcriptionist using a computer to control playback (again using a foot pedal). Dictation can also be taken over the telephone. Although some companies market this as a special product, business travelers already frequently leave lengthy voice mail messages for their secretaries to transcribe.

Dictation is used to create text documents when users refuse to use keyboards or can afford the cost of transcription as offset by the faster speed of speaking over typing or writing. Digitized dictation can also be used as a temporary information storage medium while awaiting transcription. For example, a hospital's medical records system could apply such a hybrid system. Physicians usually dictate their reports, but hospital administrators as well as medical staff prefer medical records to be stored in computers for both easy access and archival storage. Once dictation tapes are transcribed the text is easily stored on line, but until then the information is inaccessible. If the dictation were stored as digital sound files, it could be accessed immediately from a voice-and-text workstation, even as transcriptionists slowly converted the sound files to text. But for the hospital staff accessing voice in the medical record before transcription, a user interface different from the transcriptionists' foot pedal is required; voice as a document type requires a greater degree of interactivity.

Voice as a Document Type

The use of voice for announcements or in toys does not require any recording capability after the announcement has been created, while both voice messages and dictation are intended to be heard only a few times before being discarded. In contrast, voice can also be utilized as a medium for the creation of documents which are intended to be more permanent or to reach a larger audience. As a document type, voice may be used as a means of annotating text or it may itself be the primary medium. Voice documents may also include other media such as text and images. More care will be taken during recording voice documents as the stored speech is intended for more formal use than a telephone message. In many cases multiple listeners will hear the recording, especially if it is part of a presentation.

Ex.1018 / Page 88 of 343
AMAZON.COM, INC.

Computer workstation-based multimedia applications such as these have several additional requirements for stored voice not encountered in the classes of applications discussed thus far. The first requirement is a visual representation of the stored speech appearing in a document and a graphical user interface to control playback. Because of the longevity and larger audience for voice documents, more care is taken in their creation so the second requirement is a voice editor that provides a means to manipulate stored voice, including cutting, pasting, and recording new material for insertion. Editing has implications for sound file storage since it is likely to produce many small snippets of sound that result in fragmented files, and certain methods of file storage are more compatible with the internal data structures used by the editor to manipulate sound data.

VOICE IN INTERACTIVE DOCUMENTS

The remainder of this chapter will focus on uses of stored speech as a document type. As mentioned above, voice may be used alone or composed with other media to form a document. This chapter focuses on composed multimedia documents employing a display and a graphical user interface. Chapter 6 treats the problems unique to audio-only documents accessed without any display. A voice editor is also usually provided to aid authoring of voice documents. Although voice may be imported into one application from another much as text is cut and pasted between windows, discussion of this capability is postponed to Chapter 12; here we consider only document creation and retrieval.

When both text and voice are available, the different media may be used for different types of messages. Voice is appropriate for casual and transient messages, while text is preferred for messages that the recipient will likely save as reference. Voice is also desirable for its expressiveness; voice can carry subtle messages far more easily than text. It is also faster to enter messages by voice. Nicholson reported the results of a usage study of a voice and text message system in [Nicholson 1985] that supports all these observations. Chalfonte *et al.* [Chalfonte *et al.* 1991] compared voice and text as a means of commenting on text for a collaborative writing task; they found that a greater proportion of voice annotations carried "deeper" comments on style and structure, while text markings related to surface features such as choice of words and syntax.

Voice may be used to annotate text documents; it overcomes some limitations of the two-dimensional page by allowing comments to be placed anywhere with a content-associated visual icon that requires less space than handwriting. Conversely, text may be used to comment on a voice recording such as appending text remarks to a voice mail message before forwarding it to another party. Applications to support voice and text require the creation and storage of voice and a user interface to access it while viewing a document.

The Diamond project at BBN (Bolt, Beranek, and Newman) [Thomas *et al.* 1985] was an early example of incorporating voice with text. Diamond was designed to support voice and graphics in both electronic mail and more formal

Ex.1018 / Page 89 of 343
AMAZON.COM, INC.

documents. Nearly 10 years old, Diamond used LPC speech coding to minimize audio storage requirements. But the LPC coder was an external computer peripheral that was not readily available. This, combined with the low quality of LPC, limited the use of Diamond and it remained basically a prototype system. But many aspects of Diamond have reappeared in *Slate,* a multimedia mail system currently available as a product from BBN [Crowley 1991, Lison and Crowley 1989].

Slate documents can include a variety of media and Slate is designed to work in concert with a computer conferencing system, MMConf. Slate uses a commercially available speech coding card or the audio capabilities of the workstation itself. In addition to speech, Slate incorporates still images (bitmaps), geometric graphics (lines, curves, etc.), motion video, spreadsheets, and charts. Slate indicates the presence of voice in a document with a small icon that can include a line of text as shown in Figure 4.1. By default, the text contains the author's name and recording information, but the author can change the label. The reader clicks on the icon to play the associated stored voice. Slate supports a small voice editor to allow the author to modify a recording during its creation as shown in Figure 4.2.

Another example of voice annotation of text documents was Quilt at Bellcore (Bell Communications Research) [Leland, Fish, and Kraut 1988, Fish, *et al.* 1988]. Quilt was a tool to allow collaborative document creation, editing, and review. Multiple parties could add notes to the document being edited using a

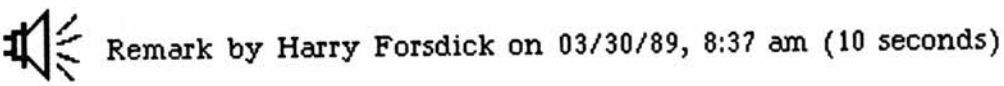


Figure 4.1. Slate shows an annotated icon to indicate speech in a document.

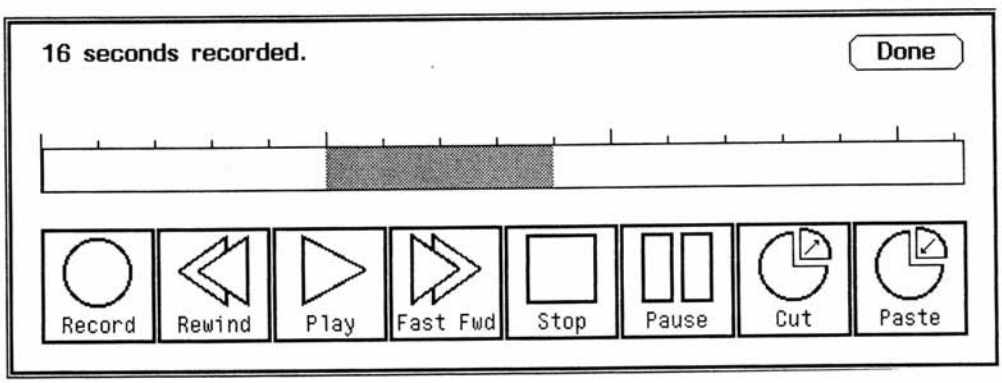


Figure 4.2. The Slate voice editor.

Ex.1018 / Page 90 of 343
AMAZON.COM, INC.

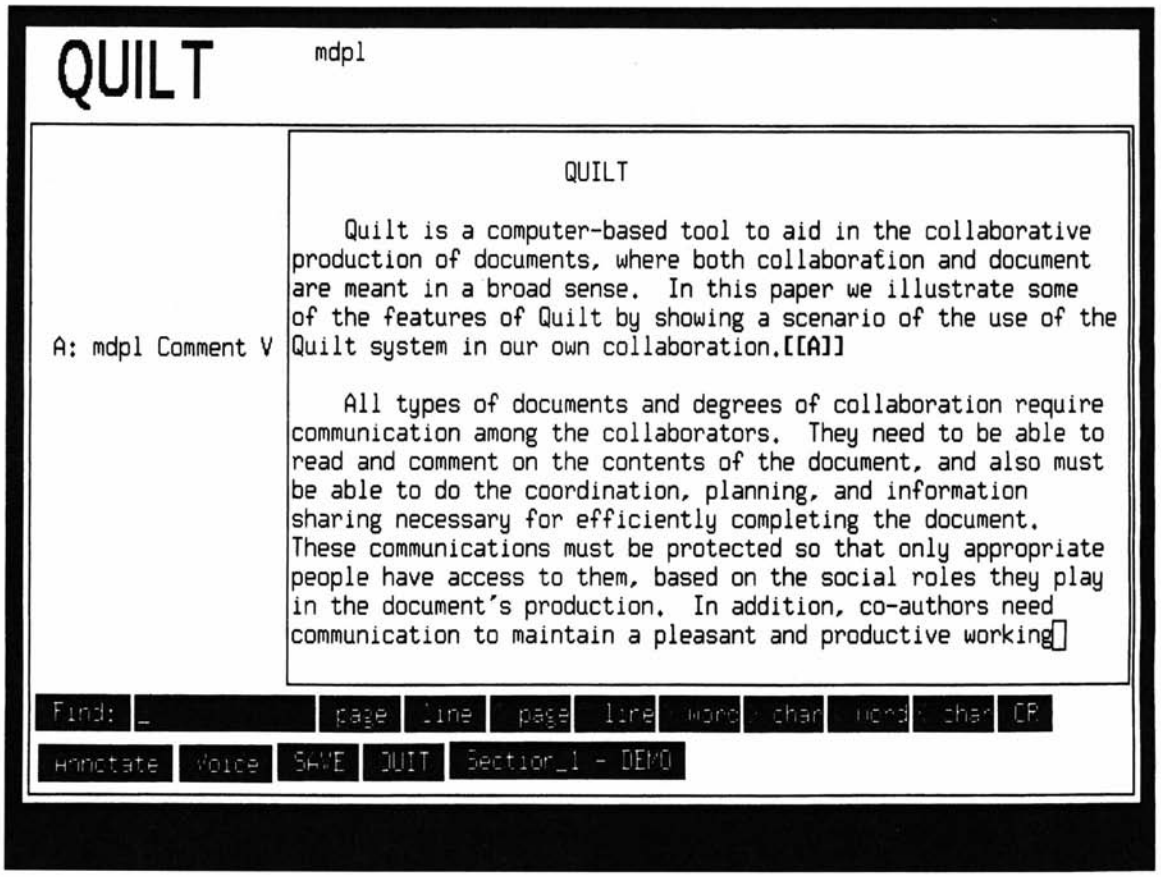


Figure 4.3. Quilt used **V** symbols to indicate the presence of voice comments in a text document.

simple voice annotation scheme. Voice recordings were indicated in the text with a "V" symbol as shown in Figure 4.3. Quilt was never widely deployed so it did not receive significant use, but it motivated preliminary studies with tape recorders that indicated that voice annotations were lengthier and more penetrating than text.

The Xerox PARC (Palo Alto Research Center) Etherphone project[1] supports voice annotation of text. It uses a "comic strip balloon" as an indicator of voice stored as part of a text document as shown in Figure 4.4. The capability to mix

[1]Etherphone is discussed in more detail in Chapter 11, and its voice editor is discussed later in this chapter.

Ex.1018 / Page 91 of 343
AMAZON.COM, INC.

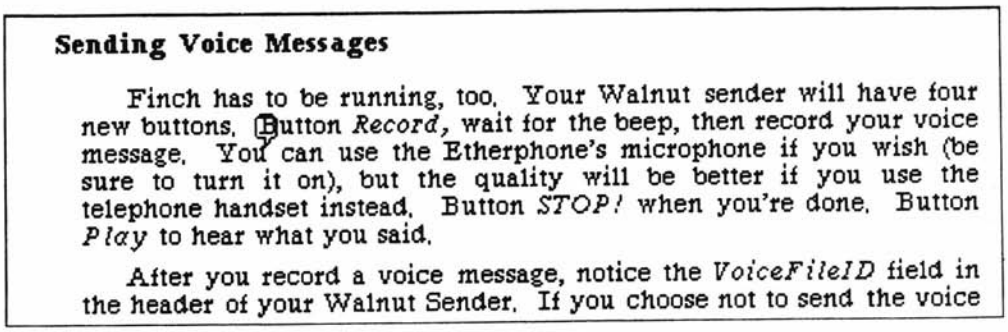


Figure 4.4. Voice is indicated by a "balloon" at Xerox PARC. (Copyright ©1986, Xerox Corporation. Reproduced with permission of Xerox Corporation.)

voice with text was extended by Zellweger to **scripted documents,** which sequentially play back (or display) portions of a multimedia document [Zellweger 1989]. A scripted document is a form of presentation with the management of time and sequence an added dimension of authoring; the script provides a single path through the document.

Stored speech is also used in multimedia documents that provide many paths for presentation under control of the "reader." For example, the MUSE system from M.I.T.'s Project Athena [Hodges *et al.* 1989] uses recorded video, audio, and textual annotation for the interactive presentation of educational material.[2] The student in a foreign language course may be asked to make decisions on the basis of a spoken dialogue played from a videodisc. Text may be employed during playback to define new vocabulary items. The student's response is indicated using the mouse, and the system responds with the next appropriate sequence.

An entirely different approach is taken in documents which consist only of voice, with nonvisual user interfaces. Muller defines an architecture for voice documents as a network of short audio segments and points out some of the timing issues with synchronizing user inputs with whichever segment is currently playing [Muller and Daniel 1990]. Resnick built a telephone-based groupware application, Hypervoice, that allowed callers to browse a community bulletin board of voice entries [Resnick 1992a]. Each entry consisted of a number of short voice recordings such as an event's time, description, and a contact person's telephone number; users navigated with touch tones. Arons' Hyperspeech used speech recognition to navigate a hypermedia database consisting of audio inter-

[2]MUSE is both the presentation system as well as the underlying language and editor used for creating the presentations.

Ex.1018 / Page 92 of 343
AMAZON.COM, INC.

views with five user interface researchers [Arons 1991b]. VoiceNotes is a user interface to a handheld digital audio storage device; users employ speech recognition and mechanical buttons to traverse and edit lists of reminders [Stifelman *et al.* 1993].

VOICE EDITING

While adding voice to a document or presentation users will occasionally make mistakes and must correct them. To change a very short voice recording it is easy to simply record the speech again; an example is the outgoing message on an answering machine. Other applications, which use longer recordings, can benefit from the addition of a voice editor. An editor allows a user to delete or move portions of a recording and to insert new audio into an existing recording; these operations are controlled with a graphical user interface. An editor can also be used to extract a portion of sound from a longer voice recording to be used in another application. The rest of this chapter considers several architecture issues for voice editors and illustrates editor user interfaces with a number of examples.

Temporal Granularity

Audio editors may provide different levels of temporal granularity for specifying the position within a sound at which a sound manipulation operation is to occur. The granularity of an editor is determined both by the application's requirements as well as by the nature of the stored sound to be edited. The temporal granularity also places requirements on the editor for display of the sound being edited.

A **fine-grained** editor allows edit operations to be performed at any point in the sound. It makes no assumptions about the content of the sound and gives the user complete control of editing. Fine-grained editors usually display sound as a waveform (see Figure 4.5) or **average magnitude** (see Figure 4.6). Average magnitude is the sum of the absolute value of some number of audio samples divided by the number of samples; samples are taken over a large enough window to average multiple pitch periods. A mechanism may be provided to zoom in and out of the sound view for local or global editing; the zooming changes the ratio of pixels to time for display purposes.

A fine-grained editor is most useful for studio work such as producing music or motion picture sound tracks. In these situations a great deal of flexibility is required as a variety of sounds may be manipulated and very careful edits are needed. For example, the Sound Droid digital audio editing system [Snell 1982] was used in the motion picture industry and emulated existing analog studio equipment both in user interface as well as basic functionality.

A **coarse-grained** editor provides much less temporal control but abstracts sound into "chunks" based on periods of speech and silence. Coarse granularity is more suitable for editing speech because speech consists of a series of semanti-

Ex.1018 / Page 93 of 343
AMAZON.COM, INC.

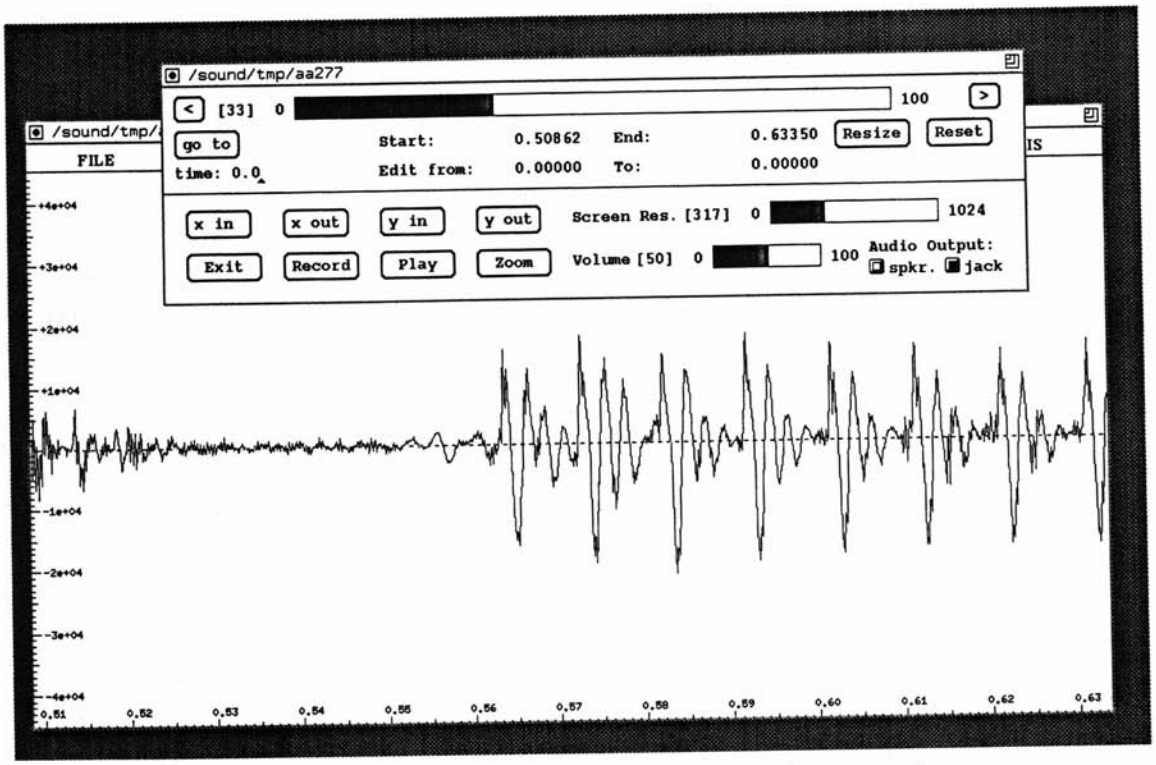


Figure 4.5. The MixView audio editor displays sound as a waveform.

cally meaningful phrases; a coarse-grained editor facilitates the user's identifying phrases and editing them as units. The abstraction of sound into speech and silence intervals also allows more succinct visual representations with reduced screen space requirements.

The coarse-grained editor is well suited to removing spurious remarks, cleaning up false starts from a recording, and replacing a sentence; it is not suited to modifying individual words. In a study of users of an early voice editor, Allen noted that subjects desired to edit individual words frequently [Allen 1983], but in practice it is nearly impossible to modify a single word in a sentence without having the result sound broken, with very unnatural prosody. Since users will not be able to perform successful fine-grained editing of voice, coarse-grained editors make no attempt to support this function, and use a simple binary representation to simplify the user interface.

Manipulation of Audio Data

An audio editor must manipulate stored sound data in response to the user's commands. As users are likely to make changes and review them several times before achieving the desired result, it is useful if the editor allows preview before com-

Ex.1018 / Page 94 of 343
AMAZON.COM, INC.

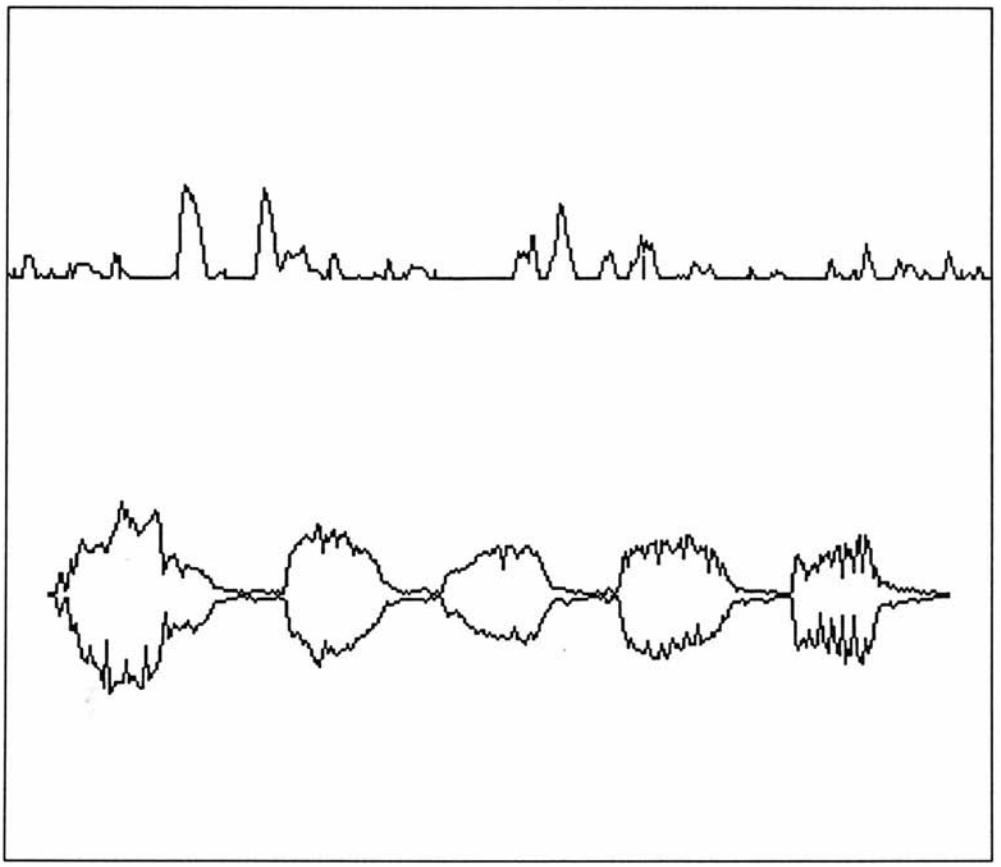


Figure 4.6. Average magnitude display of a sound (top) and an envelope display showing both positive and negative peaks (bottom).

mitting the changes to the original sound file. Although this could be done simply by editing a copy of the sound file, most editors instead operate on an internal representation of the sound that allows users to edit and review without changing the file at all. When finished the user may choose to save the edited sound at which point the file is changed. Many text editors employ similar techniques.

More importantly, editing an in-memory representation of the file offers significant performance improvements over editing a file directly. For example, if the user deletes the first character of a text file or the first 100 milliseconds of an audio file, it is time consuming to move the entire remainder of the file.[3]

[3]Under some operating systems this requires copying the entire file as well.

Ex.1018 / Page 95 of 343
AMAZON.COM, INC.

Instead, the editor just modifies its description of the sound to note that the very beginning of the file should be skipped over during playback. At the end of the editing session the disk file can be changed in a single operation avoiding slow file operations during editing. This is even more important in the editing of sound than in text as sound files are much longer than most text files. A chapter of this book may takes roughly 40,000 bytes of text storage, but for many speech coding algorithms this would represent less than 10 seconds of sound.

An audio editor may represent a sound being edited as a **buffer** or **playback list.** This can be implemented as a linked list where each entry in the list points to a sound file and indicates start and end points. Figure 4.7 shows how such a representation is employed during editing. When a file is initially read into a buffer (shown in the top portion of the figure), the buffer includes the entire sound. The edit list contains a single entry with a start point of zero and stop point at the end of the file; duration is represented by shading in the figure. A "cut" operation removes a portion of the sound (shown in the middle of the figure). Now the buffer list has two entries. The first entry points to the initial part of the file at offset zero and extending up to the beginning of the cut section. The second buffer entry points to the same file but starts after the cut section and extends to the end of the sound. When the edit is reviewed, the editor plays the first chunk of the file and immediately skips the middle part of the file to play only the final segment. When the buffer is then written out to a file, a single homogeneous file is created by merging the two sections of sound (shown in the bottom of the figure).

To insert a new sound into the middle of a buffer, the buffer must be split into two list entries with the new sound added to the list in between them as shown

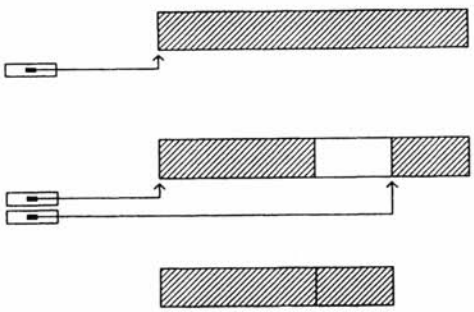


Figure 4.7. Edit buffers. At the top, a single buffer node points to an entire file. In the middle, after removing a portion of the sound, two buffer nodes are required to point to the two segments of the file. At the bottom is the correct representation after the buffer has been written to a file; the two portions of the original file have been merged into a single continuous file.

Ex.1018 / Page 96 of 343
AMAZON.COM, INC.

in Figure 4.8. With each edit operation performed by the user, one or more entries are likely to be added to the buffer list. After a number of operations, the resulting representation is complex (see Figure 4.9), but it is hidden completely from the user. Depending on how sound is stored in disk files, the complexity may be largely hidden from the editing program as well; some sound file systems

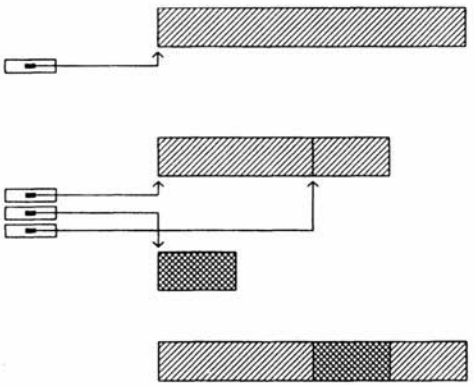


Figure 4.8. Edit buffers showing insertion of a new sound into an existing sound. The original sound at the top must be divided into two segments, each with its own buffer node and a new node inserted in their midst. The resulting buffer in the middle may then be written to a single file at bottom.

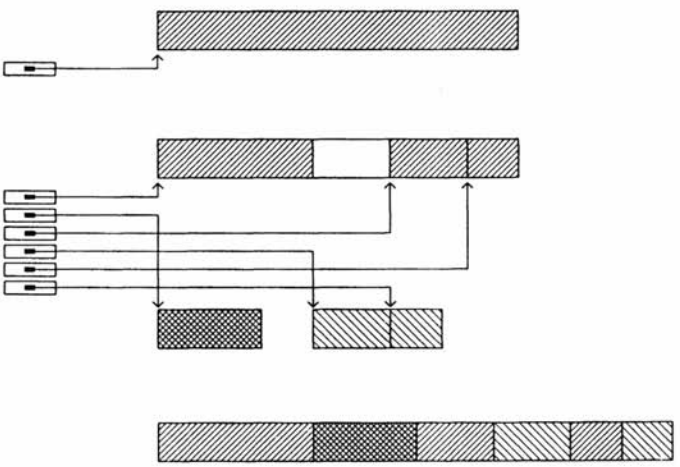


Figure 4.9. After a number of edit operations, the resulting buffer may become rather complicated.

provide transparent sequential access to chunks of sound stored in different locations on the disk.

EXAMPLES OF VOICE EDITORS

All sound editors include a user interface to the editing functions. These functions include selection of a sound file for editing, playback of the sound, deleting part of the sound, and inserting new sound. The appearance of the user interface depends on the window system, graphical toolkits, and style conventions for the particular operating environment in which it runs. The editor may be modeled after existing tools with which users are familiar, such as text editors for the office and audio equipment for the studio.

One of the earliest voice editors was built by Maxemchuk at Bell Laboratories as part of an experimental speech storage facility [Maxemchuk 1980]. This editor had a primitive interface based on an alphanumeric terminal; a subsequent version used a more sophisticated display. This section briefly describes a number of more recent audio editors; they exhibit a wide range of user interfaces and sound representations yet share many aspects of underlying software architecture.

Intelligent Ear, M.I.T.

The Intelligent Ear was an early voice editor from the Architecture Machine Group (M.I.T.) [Schmandt 1981]. It displayed a sound as an amplitude envelope using average magnitude to modulate both height and brightness as shown in Figure 4.10. The entire sound was displayed on the screen; long sounds were shown with less temporal detail than short sounds because of the compression required to fit them on the screen. The Intelligent Ear also used speech recognition to attempt to detect keywords in the sound and displayed these as text with brightness indicating the degree of certainty with which the word was recognized.

This editor used a touchscreen for input. When the user touched a location on the sound display, a cursor moved to that location. When the *play* button was touched, playback began at that point and the sound display changed color while the cursor moved in synchronization with playback. This motion was provided to aid the user's mental mapping of the visual cue of average magnitude to the actual contents of the sound. Other editing functions included *delete* and *record,* which enabled the user to specify an insertion point at which to record additional speech. Edit operations were buffered in that they did not change any sound file data, but this buffering was only one operation deep. After an insertion or deletion the user could play through and listen to the result; only after accepting this result was the sound file actually changed on disk. An edit operation was either accepted or rejected by touching a red or green button which appeared after the edit, but the user was not able to make multiple edits and preview the result without modifying the file.

Ex.1018 / Page 98 of 343
AMAZON.COM, INC.

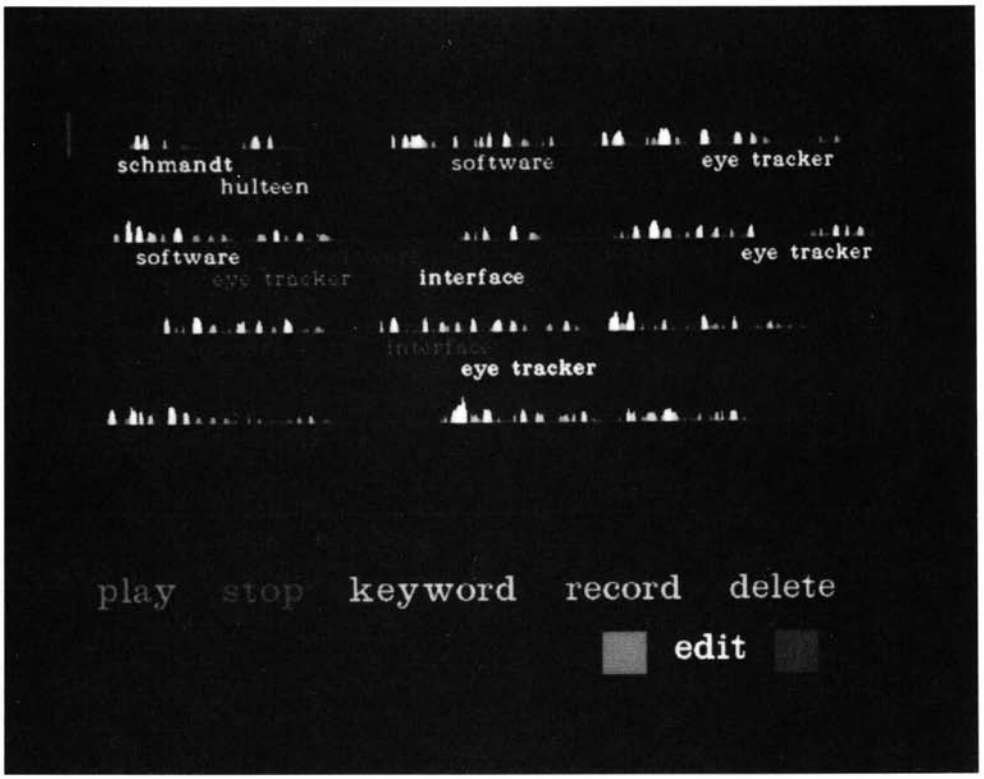


Figure 4.10. The touch-sensitive user interface to the Intelligent Ear editor.

The Intelligent Ear was a demonstration project which saw only limited internal use by its authors to edit voice prompts in later research projects. It was novel in its early use of a color graphical interface and touchscreen for voice editing.

Tioga Voice, Xerox PARC

The Tioga Voice audio editor at Xerox PARC was modeled after the Tioga text editor's user interface and underlying data representations with the intention of making editing voice similar to editing text [Ades and Swinehart 1986]. Tioga Voice was designed to support audio in a variety of applications and could be used as a listening tool as well as an editor.

The Tioga Voice user interface shown in Figure 4.11 supported a number of features. Portions of speech and silence were displayed by a binary representation in a small bar (or "capillary tube" as described by its creators), and text annotation could be added as well. On a color screen, recent changes or additions appeared brighter than older sound segments. During recording the display changed dynamically showing the new sound as a string of arrows both to keep a sense of

Ex.1018 / Page 99 of 343
AMAZON.COM, INC.

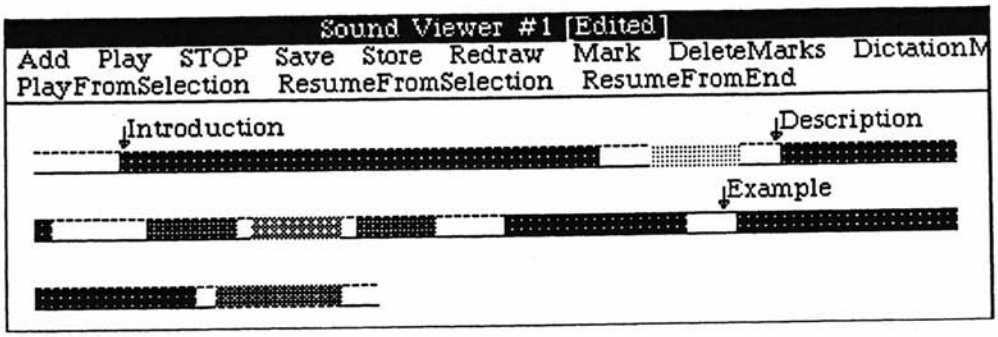


Figure 4.11. The Tioga Voice editor. (Copyright ©1986, Xerox Corporation. Reproduced with permission of Xerox Corporation.)

its relationship to the whole recording as well as to remind the user that the recording was still in progress. During playback, a marker moved across the sound bars indicating the current play position.

To facilitate the manipulation of sound files, Tioga Voice kept track of the current *selection,* i.e., the piece of sound most recently referenced. During dictation mode, the selection corresponded to the most recent portion of speech recorded to allow for quick review or cancelling erroneous input. If the user began recording, coughed, and then stopped recording, this audio segment could be deleted by a single mouse click.

Tioga Voice was built on top of the Etherphone system, which provided sound storage and voice services through server processes running on other computers. Although voice segments were stored as sequential files on disk, they were accessed through the file server as a list of segments referred to as a *voice rope* [Terry and Swinehart 1988]; text was stored in a similar fashion, also referred to as a *rope*. A complete sound consisted of the rope database plus raw sound storage (see Figure 4.12). The voice ropes were similar to the edit buffer schemes described above except that they were saved on disk as part of the storage mechanism provided by the server for all sound access. Each rope represented a path through portions of one or more sound files; editing simply created more ropes. Because editing simply created more ropes instead of actually modifying the underlying sound files, periodic garbage collection was required to recover unreferenced disk files.

PX Editor, Bell Northern Research

Another example of a sound editor is that which was developed for the PX (Personal eXchange) project at Bell Northern Research's Computer Systems Research Laboratory [Kamel, Emami and Eckert 1990, Bowles *et al.* 1991]. PX is discussed in greater detail in Chapter 11; in this section we consider only its sound editor.

The PX editor, shown in Figure 4.13, ran on Apple Macintosh computers with an external hardware interface for speech input. This editor supported several

Ex.1018 / Page 100 of 343
AMAZON.COM, INC.

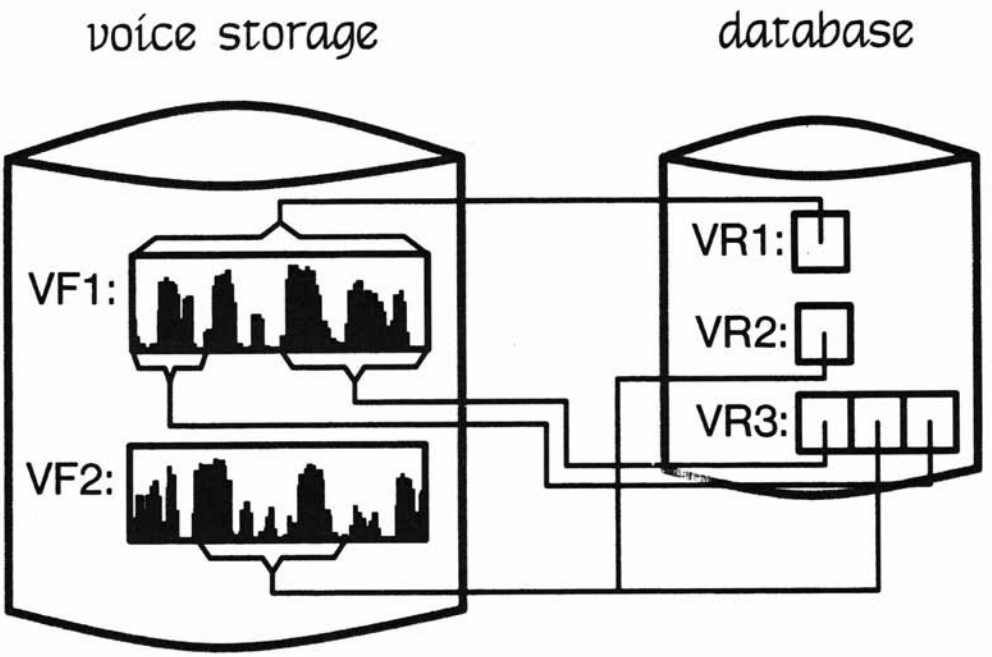


Figure 4.12. Voice ropes, showing the rope database which references portions of linear sound files. (Reprinted with permission from *Transactions on Computer Systems* ©1988, ACM.)

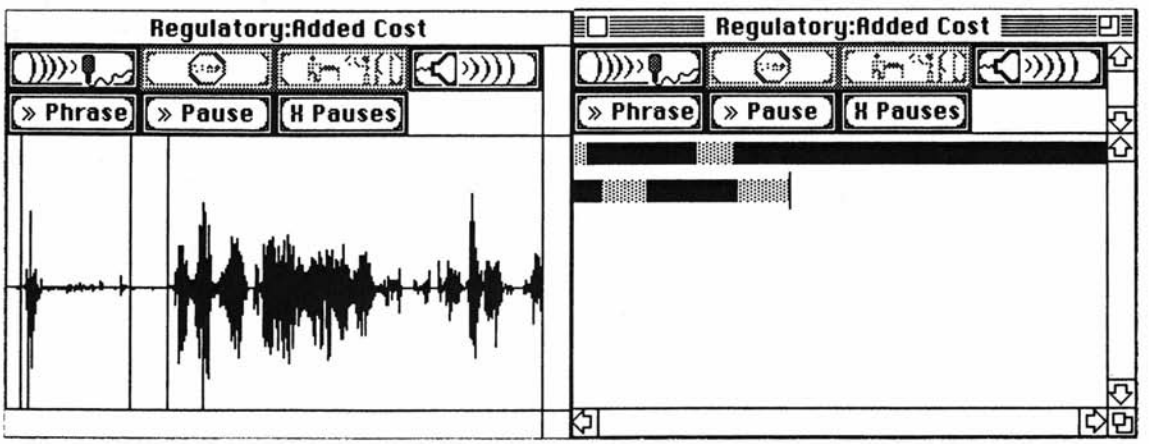


Figure 4.13. The PX sound editor showing two sound representations. The left view is a waveform envelope, while the right view shows speech and silence intervals. (©1990. Reprinted by permission of BNR.)

Ex.1018 / Page 101 of 343
AMAZON.COM, INC.

representations of sound. Figure 4.13 shows the same sound displayed in two windows: one using a waveform envelope and the other showing speech and silence intervals much like the Tioga Voice editor. This simple editor was meant to be a common tool used in a variety of applications such as creating a voice mail greeting and annotating text. In addition to the playback and recording functions described previously in this chapter, the PX editor allowed the user to skip to the next phrase during rapid playback.

PX utilized small segments of sound file storage called "voice clips," which were stored and accessed much like Xerox's voice ropes. Unlike Etherphone, however, the PX editor was implemented entirely on a single computer; there was no server process involved. Voice clips described a series of portions of a file to play in sequence, and every edit operation could produce multiple new voice clips. A sound file held a number of voice clips followed by the reference audio data, which was played as indicated by the clips. In contrast to Xerox's voice ropes, the voice clip database was stored in the same file as the raw audio so it could refer only to data in that file. If a new sound was to be made by merging two other sounds, these would have to be copied into the data area of the new sound file.

Sedit, Olivetti Research Center, and M.I.T. Media Laboratory

A similar audio editor, sedit, was originally written at Olivetti Research Center as part of the VOX audio server project [Arons *et al.* 1989].[4] It was subsequently extended substantially at the Media Laboratory where it is in use today. Sedit uses the X Window System for its graphical user interface and employs a sound editor widget.[5] The sound editor widget (see Figure 4.14) uses black and white bars to display periods of speech and silence; this display is similar to that used by Tioga Voice. A scroll bar beneath indicates how much of the entire sound file is visible and allows the user to choose a portion of the sound to view in detail. As the sound plays, a cursor (a vertical bar) moves across the display synchronously with playback; if necessary, the scroll bar also moves to keep the current location in view.

Sedit can edit multiple sounds simultaneously allowing data to be cut and pasted between sound files; Figure 4.14 shows several sounds active within the editor. The horizontal lines in the figure indicate the portion of a buffer selected for the next operation. Selection is performed by dragging the cursor across the sound with the mouse. Once a region is selected, the edit operation is indicated by clicking on the appropriate button.

[4]VOX is described in detail in Chapter 12; discussion here is limited to support for editing.

[5]In X Windows a "widget" is a basic user interaction object such as a button or scroll bar that can be instantiated multiple times in multiple applications.

Ex.1018 / Page 102 of 343
AMAZON.COM, INC.

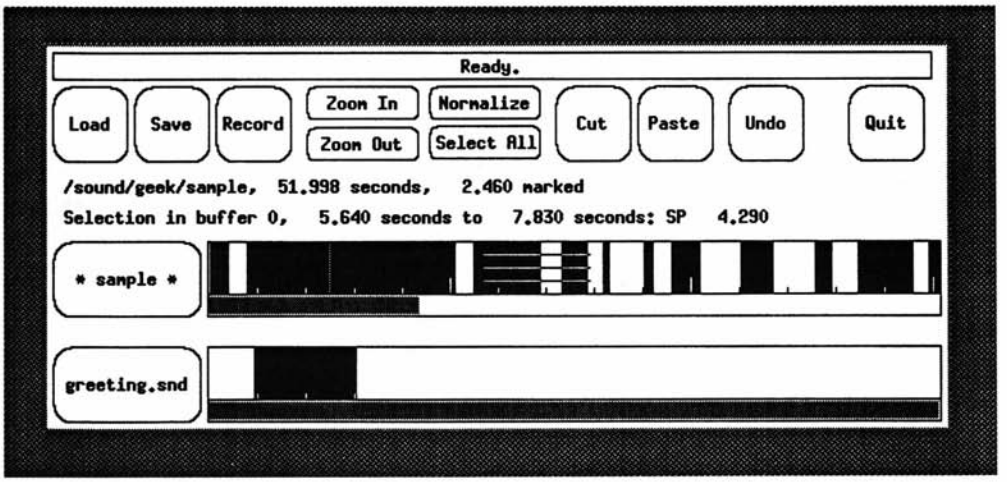


Figure 4.14. The *sedit* audio editor.

A year after the demise of the VOX project, sedit was revived at the MIT Media Lab. Capabilities that formerly had been implemented in the server were included in the editor and new functionality was added. One new function normalizes audio level among all buffer segments; this is useful when editing together source material recorded with different microphone gain levels. Other functions allow changing the time scale of the sound display widget and interaction with a selection mechanism allowing sound to be moved between applications (see Chapter 12). This is the version of sedit shown in Figure 4.14.

Pitchtool, M.I.T. Media Laboratory

Pitchtool was a special-purpose editor written at the Media Laboratory to aid in the study of intonation. It is mentioned here as an example of an editor interface designed specifically for a particular application. Pitchtool was used as much to analyze recordings as to change them and so included such capabilities as variable playback speed, textual annotations, and spatial scaling of the sound to show extended portions of dialog or detailed views of individual phrases.

Pitchtool, shown in Figure 4.15, displayed pitch, energy, and segment duration of a digitized speech file. In the figure, the solid line represents pitch, while the broken line represents energy. Pitchtool also provided for modification of these parameters to allow hand-editing of the pitch and energy for experiments in the synthesis of intonation. The mouse could be used to "paint" a new pitch track or two points could be selected with pitch interpolated linearly between them.

Because it was designed to manipulate the pitch of sounds, Pitchtool used Linear Predictive Coding (see Chapter 3) for its speech storage. LPC encoding was

Ex.1018 / Page 103 of 343
AMAZON.COM, INC.

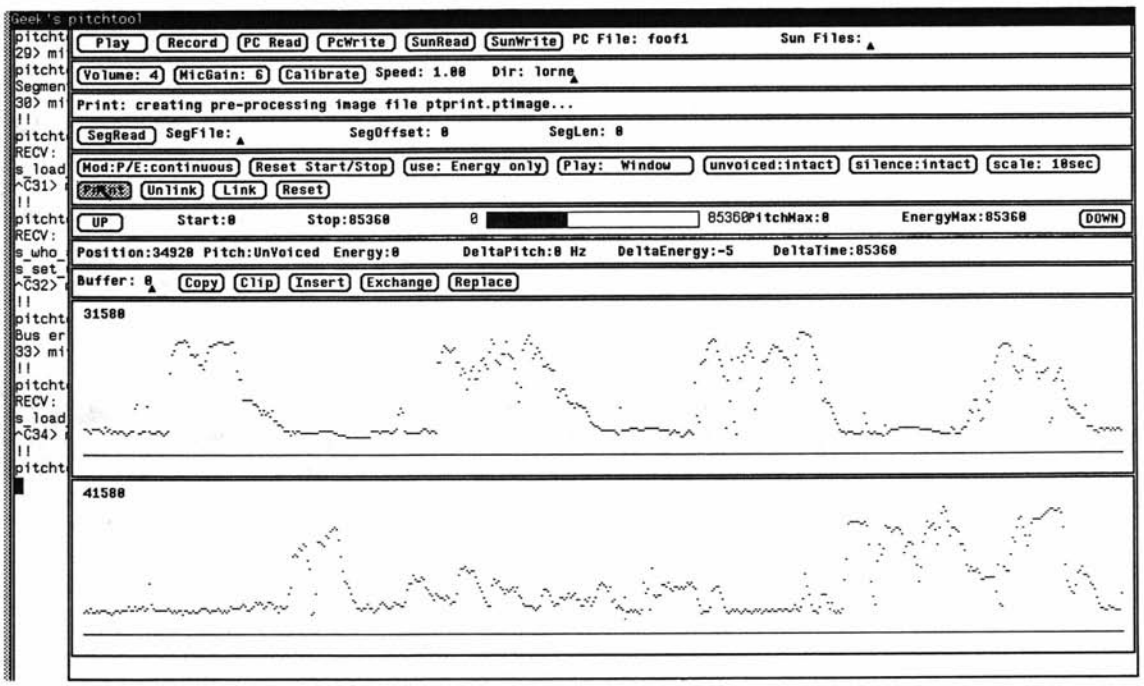


Figure 4.15. Pitchtool allowed editing of pitch and energy.

performed by a server, which included a signal processing card and small computer with local storage, communicating to the host workstation via a serial interface (this server is described in more detail in Chapter 12). As with the original sedit editor, all manipulation of sound data was implemented in the server; Pitchtool merely provided the user interface.

SUMMARY

This chapter described classes of applications for stored speech with emphasis on the underlying system and coder requirements for each. For some of these classes, speech is played only while others require recording as well. Recordings can be casual as in voice mail messages or more formal for presentations or audio memoranda. Recording for dictation may be short lived before it is converted to text. Each class of operation places different requirements on audio storage in terms of the quality required for data compression algorithms, speed to access voice data, and the synchronization of playback with a graphical user interface or other presentation media.

Particular emphasis was placed on interactive audio documents because they highlight requirements for screen-based user interfaces and sound editors. The

Ex.1018 / Page 104 of 343
AMAZON.COM, INC.

range of example systems and editors has revealed a variety of graphical user interfaces and representations of sounds. Editors generally employ in-memory buffers to describe sound file segments during editing, but this capability may be included in the set of primitives provided by the sound file management system. Similar storage issues arise with file formats and editors for all multimedia data that require very large files that may become fragmented as a result of editing operations.

Ex.1018 / Page 105 of 343
AMAZON.COM, INC.

5

Speech Synthesis

The previous two chapters discussed techniques for digital encoding of speech and how different classes of applications utilize this recorded speech. In some instances, such as a voice mail system, the computer treats recording as opaque data; it captures what one person says and relays it to another without knowledge of its lexical content. In other applications, such as telephone-based access to flight schedules or announcements of the approaching station on a subway, the application knows the content of prerecorded snippets of speech and pieces them together to generate spoken output. But although the computer "talks" in such applications, its repertoire is limited to a few prerecorded phrases. Speech synthesis is the process of generating speech from text; for any string of words a speech synthesizer can approximate the way a human would read these same words.

Although synthesized speech cannot be expected to achieve the quality of the human speech that it mimics, the flexibility of synthesis makes it extremely useful for a variety of applications. Synthesis allows voice output in discourse systems in which the computer has a wide variety of things to say, and it can easily be driven by language generating programs that produce text. Synthesis is helpful for voice access to databases because databases may have a very large number of entries to pronounce (such as names of people, streets, or businesses). Finally for some applications the computer may need to speak human-authored text such as an electronic mail message or a text file.

Transforming text into speech is done by a process called **text-to-speech synthesis** or **synthesis-by-rule.** These terms are chosen to contrast some speech coding methods such as Linear Predictive Coding (see Chapter 3), which

Ex.1018 / Page 106 of 343
AMAZON.COM, INC.

are sometimes called **analysis-synthesis** algorithms. Such coders first analyze the speech in terms of parameters to control an underlying vocal tract model, and then "resynthesize" it for playback through a digital system that implements this model. Speech coders can only store spoken speech; speech synthesizers accept arbitrary text as input and generate speech. Throughout this book, the term "speech synthesis" is used to refer exclusively to text-to-speech generation.

To begin to appreciate the difficulty of synthesizing speech, let us consider an apparently simple approach to synthesis that is actually impractical although quite suitable as a substitute to synthesis in certain limited situations. Suppose we simply record a person speaking every possible word and then string these audio segments together for speech output? This approach is impractical for several reasons.

First, there are simply too many words. Not only do we need to digitize the root forms of nouns and verbs, we also need their forms in combinations. For "play," for example, we might want "plays," "playful," "playing," "played," etc. or compounds such as "horseplay."[1] Even if we somehow coped with storing all these word forms, it would be even less practical to store all the proper nouns that might be required to speak from some databases [Spiegel 1985]. Finally, languages change and new words and acronyms ("DRAM," "ISDN," "downsize") keep appearing in English or any language, necessitating recurring updates of the recorded speech.

Second, even if we *could* store every word which might be spoken, they would sound very awkward when strung together. Words change pronunciation in a spoken context. Sentences have a melody and rhythm that changes the pitch and duration of syllables. Phonemes at the beginnings and endings of words are spoken with variations that harmonize with surrounding words. A word spoken in isolation is said to be in **citation form,** but this is not how we speak in normal conversation. Consider the word "the"; when asked to say this word by itself, many people instinctively pronounce it as "thee," but we actually speak it this way only when the following word begins with a vowel. Many words exhibit this phenomenon. "To" is usually spoken more like "ta," and the vowel in "for" practically disappears in fluent speech.

Although these observations are meant to argue that conversion of text to speech is more difficult than patching together digitized words, in some situations it may suffice to record a number of short phrases and concatenate them; this technique is used in many telephone-based applications. For example, the caller speaks with a human directory assistance operator, but then a computer voice recites the number. What actually transpires is that a number of short recorded segments are played starting with "The number is . . ." followed by a separate recording of each digit.

Although the phone number is spoken by concatenating recordings of each digit, some sophistication is required to make the number sound natural. All dig-

[1]Note that we know not to pronounce the "e" in such a compound word.

Ex.1018 / Page 107 of 343
AMAZON.COM, INC.

its in a telephone number are not pronounced the same; intonation is used to group them. North American numbers consist of three digits (area code), three digits (exchange), and four more digits. Each group is pronounced separately, with a falling pitch indicating phrasing (see Figure 5.1). The last digit in each of the first two groups has a rising pitch, which serves as a continuation marker or cue that more information will follow the pause. For each digit, three (or sometimes four) recordings are made, and the appropriate one is chosen depending on whether the digit is in the initial, medial, or terminal position in the number.

SYNTHESIZING SPEECH FROM TEXT

How can text be transformed into speech? As just discussed, it is impractical to simply record each word because of pronunciation problems as well as the amount of storage required for the lexicon. We cannot go directly from text to sound by pasting words together; instead, some smaller unit of representation is required for sound generation. For efficient synthesis, this unit should be significant either from the perspective of the written language (words or letters) or from the spoken language (syllables or phonemes).

English is a difficult language to synthesize largely because its orthography (the written version of the language) is highly irregular; more specifically, the mapping from letters to sounds is not one to one. The same sound may be produced from various letters (e.g., the initial phoneme in "kid" and "cat"). Worse still, the same letters can be pronounced in different ways (e.g., the final four letters in "tough" or "through").

In some languages, the stronger relationship among letters and sounds would suggest that letters be used as a unit of synthesis. An example is the Turkish language; the Turks replaced their former Arabic script with a Roman alphabet earlier this century as part of the "modernization" of the country. The new alphabet was created specifically for the language from symbols used in other languages,

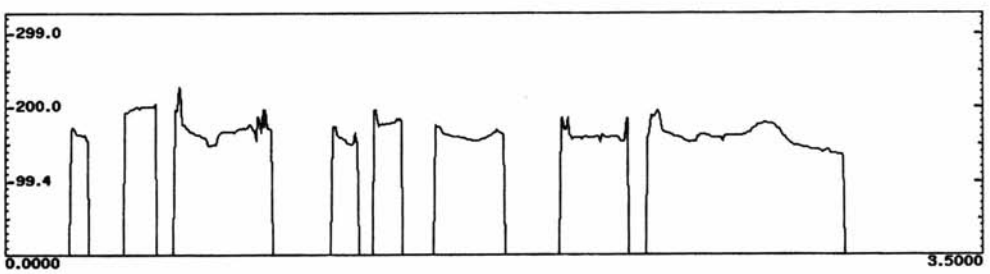


Figure 5.1. Pitch track of a telephone number including area code. Note the phrasing of the three groups of numbers. The first two number groups show a small terminal rise, which is a cue to the listener that there is more to come. There is no such rise in the final number group. The vertical axis is pitch in Hertz, the horizontal axis shows time in seconds.

Ex.1018 / Page 108 of 343
AMAZON.COM, INC.

and it is conveniently phonetic. Equally important, spoken languages continually change but only slowly. So spoken Turkish has not yet diverged from its recently specified written form. In the new Turkish alphabet, not all the letter symbols are recognizable in any single European language; for example, both "o" and "u" exist in an umlaut and nonumlaut form as in the German "u" and "ü," but "i" also has a dotted and undotted form ("ı" and "i"), which German does not. However, once the single pronunciation of each letter is learned, it is easy for someone who does not speak Turkish to read it aloud. This is exactly the task of a speech synthesizer.

Since English is not this simple, it is necessary to first convert the letters to a less ambiguous representation and then apply sound realization rules to that representation. The most common alternative form to the letter is the phoneme as there are a fairly small number of phonemes in any language, simplifying the sound generation rules.[2] Synthesis is composed of two steps; the first converts the text to a string of phonemes (with intonational markers), and the second realizes the output of the first as a speech waveform. This is depicted in Figure 5.2. Note that in the Turkish example, since each letter maps to exactly one phoneme, we can simply skip the first step in this model except that intonational markers must still be added to each syllable.

FROM TEXT TO PHONEMES

So how can text, an ordered series of letters, be converted to the equivalent string of phonemes? Two general approaches may be identified: a **pronunciation lexicon** (dictionary) or a set of **rules** similar to what we learn in elementary school. Both are useful and a hybrid approach is optimal.

The dictionary approach is simple. A string of phonemes is stored for each word, and conversion of text to phonemes is accomplished by lookup. Several problems prevent a lexicon-only solution, however. First, the dictionary grows rapidly as a result of the sizable vocabularies required for many applications. Second, the dictionary must be stored in nonvolatile memory in a stand-alone synthesizer. Third, at some point in the dictionary lookup a "morphological decomposition" analysis of the text must occur, if only to identify simple forms such as plurals and past tense; if both "cat" and "cats" are to exist as distinct

[2]The number of phonemes in a language varies from a low of 13 (Hawaiian) to a high of about 75. English is in the middle with approximately 40 phonemes.

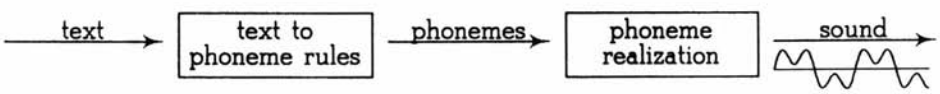


Figure 5.2. A basic model of speech synthesis.

Ex.1018 / Page 109 of 343
AMAZON.COM, INC.

entries, the dictionary must be much larger. Fourth, sometimes correct pronunciation cannot be gained from looking at a word in isolation, but rather, sentence context must be employed ("read" has two pronunciations depending on tense; some other examples are discussed below on the topic of lexical stress). Finally, no matter how large the dictionary is, there will always be words it does not contain such as proper nouns, jargon, and new words.

The rule-based approach, in contrast, uses knowledge of spelling rules to derive pronunciation from the text. There are several advantages to such techniques. First, the set of rules is much more concise than a lexicon of pronunciations for each word. Second, it is more flexible; effective rules can generate a plausible pronunciation for almost any word including names. Third, rules represent succinct knowledge about a language. Rules may be portable across languages, and the rules for a language can generate pronunciations for names from that nationality even while synthesizing a different language.

Although a completely rule-based synthesizer is attractive, this is quite difficult and not the easiest path to the goal of accurate synthesis of commonly used words. Since a dictionary may more easily embody correct pronunciation than some of the more general spelling rules, the usual approach is to store common words in the lexicon and apply rules to the remaining words. It is best to put in the lexicon only those common words that are known *not* to be pronounced well by the rules. If lookup fails, a rule is invoked which is either correct (or else the word would be in the lexicon) or at least a reasonable guess for an unlisted and probably unusual word (which would not be in the lexicon under any circumstances).

Even for a completely rule-based synthesizer, it is quite useful to include a user-definable phonetic exceptions dictionary to deal with important words that are not pronounced correctly such as names or acronyms common in a particular organization. Although these words may be uncommon in the language as a whole, local use may be frequent; if they are mispronounced, the result will be greater listener dissatisfaction.

The process of text-to-phoneme translation is performed in several steps. The first step is text preprocessing or **normalization** to convert symbols and abbreviations into their full-text spellings such as converting the symbol "$" to "dollar." Normalization is not quite as simple as it may first appear; the best full-text spelling of "$4.35" is "four dollars and thirty-five cents," but "$2.00" is simply "two dollars." Similarly, some abbreviations, such as "Dr." and "St.," can represent one of several possible words. "Dr." is usually "doctor" when the word following is a proper noun ("Dr. Jones"); it is pronounced "drive" when a proper noun precedes it ("Jones Dr.").

The next step is **morphological analysis,** which copes with words composed of several root parts (morphemes) including plurals and compound words such as "baseball" and "backpack." Many words in English are built by combining root forms with **affixes.** An affix may be a prefix such as "un" in "unusual," or a suffix such as the "ing" in "buying." A word may have multiple affixes at either end, such as "usability," from "use" + "able" + "y." This last example also shows how spelling may change as affixes are added; these regular spelling changes must be included in the morphological decomposition.

Ex.1018 / Page 110 of 343
AMAZON.COM, INC.

Since words are built from morphemes, there will be far fewer morphemes than words, making storage in a lexicon more practical. Text is broken down into morphological units, pronunciation is looked up for each, and rules are applied to those units for which lookup fails. This process of producing phonemes from text is summarized in Figure 5.3.

Additional Factors for Pronunciation

Unfortunately, simply converting text to a stream of phonemes is inadequate for the next stage of synthesis: generating sound. Several confounding factors complicate the model shown in Figure 5.2; additional information, such as **lexical stress, coarticulation,** and **intonation** must accompany the phonemes. Lexical stress is the pattern of syllabic emphasis within a word. Coarticulation is the change in pronunciation of phonemes as a function of their phonemic environment, i.e., the sounds preceding and succeeding them. And overall sentence intonation, or **prosody,** requires adjustments to the output of the text-to-phoneme processing.

Lexical Stress

In English, as in many languages, not all syllables are created equal. We learned in elementary school that every word has one syllable carrying primary stress and possibly others carrying secondary stress. How do we generate stress acoustically? Although the intuitive answer is the volume (amplitude) of the speech, this is only a secondary factor. As Fry demonstrated in some key experiments [Fry 1958], stress is carried primarily as variations in pitch and duration of the appropriate syllables. A stressed syllable is somewhat longer than an unstressed syllable and will also carry higher (or lower) pitch than it would otherwise as shown in Figure 5.4.

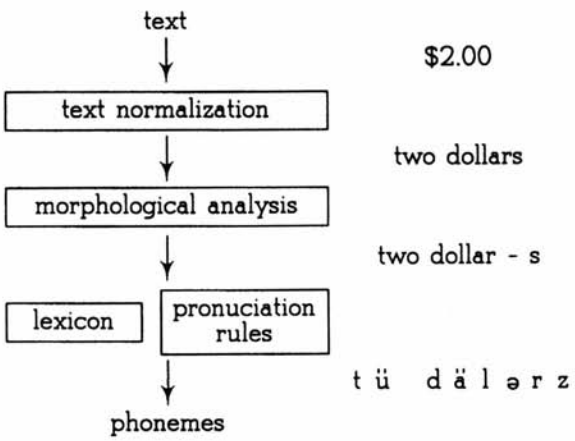


Figure 5.3. Steps in converting text to phonemes.

Ex.1018 / Page 111 of 343
AMAZON.COM, INC.

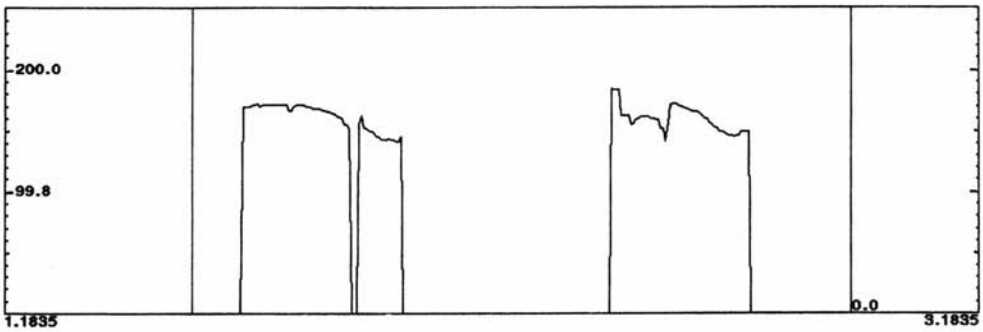


Figure 5.4. Pitch tracks showing lexical stress. The word on the left is CON-duct. The word on the right is con-DUCT. Note that the stressed syllable is longer and higher pitched.

Lexical stress is part of a complex interplay between the intonation and rhythm of a whole sentence. Discussion of lexical stress is limited to that which can be detected in a word spoken in isolation, in its citation form. Stress covers many phenomena, all realized as variations in rhythm and meter; lexical stress is what differentiates syllabic "accent" as opposed to emphasizing one word over another in a sentence. Lexical stress is important for understanding spoken language; incorrectly assigned stress may make a word unintelligible. For some English words the assignment of stress differentiates noun and verb forms, e.g., "convert" and "conflict" (see Figure 5.4). Determining which syllable to stress requires a reliable syntactic analysis of the sentence.

Stress confounds text-to-phoneme conversion because stress is not directly phonemic. In addition to producing the proper phonemes, the syllables containing those phonemes also must be marked for stress. This is done by assigning pitch and duration for each voiced phoneme during the conversion to phonemes.

Coarticulation

Coarticulation is the process whereby the pronunciation of a phoneme changes as a function of its surrounding phonemes. This change may be an allophonic variation, that is, substitution of an acoustically distinct version of the same phoneme. An example of this is the "t" sound in "but" versus "butler"; terminal stop consonants tend to be released. Alternatively, the substitution may result in an entirely different phoneme from the citation form. Consider the second "d" in "Did you . . . ," which can be pronounced much closer to a "j" without loss of intelligibility.

Coarticulation occurs because the vocal tract is a physical system. The articulators (tongue, lips, etc.) have mass and make continuous motions that are not instantaneous, due to inertia and the physical processes controlling the various muscle groups. Although for each phoneme there is a desired target configuration for the articulators (such as tongue touching the palate), in practice the articula-

Ex.1018 / Page 112 of 343
AMAZON.COM, INC.

tors may never fully realize that configuration. Instead, they simply make a gesture in its direction, which is sufficient to change the sound emanating from the vocal tract just enough to cue the listener to the identity of the intended phoneme. Pronunciation is sloppy. This is not a defect in speech; rather it is an accommodation to the speech production process that results in an ability to communicate faster than we could otherwise.

In some cases these coarticulation effects are localized within a word. The consonant "r" is a good example; the sound of "r" is realized largely through modification of the preceding vowel. In other cases these effects occur across word boundaries. Consider the fricatives in "the cats went" and "the cats sat"; the pause required to distinguish the plural form of cats in the latter introduces more aspiration on the first "s." Speech synthesized without considering these coarticulation effects will be stilted and perhaps less intelligible. Fortunately, many coarticulatory effects can be predicted as they follow rules [Oshika *et al.* 1975], although a complete set of such rules has not yet been compiled for English.

There is also interaction between coarticulation and lexical stress; in general, the unstressed syllables are much more subject to change. An example is **vowel reduction,** in which the vowel of an unstressed syllable is shortened and turned into a schwa.[3] In the extreme, the reduced vowel may simply vanish; consider how many syllables are pronounced in "chocolate" (two or three?). Note what happens to the "e" between "manager" and "managerial"; in "managerial" the "e" is stressed and hence not reduced.

Intonation

A final consideration, before phonemes can be realized as sounds, is the overall melody or intonation of the sentence. Intonation refers to the pitch contour of the sentence, both in terms of which words receive greater emphasis as well as the general slope (rising or falling) of the pitch contour. Intonation differentiates questions from statements even when the words themselves may be identical, and it also reveals which words in a sentence are particularly emphasized.

Consider the intonation of the sentences "She went to Paris." (simple statement of fact) and "She went to Paris?" (expression of disbelief, or request for clarification). The first sentence (see Figure 5.5) has overall falling pitch typical of simple declaratives. The second sentence (see Figure 5.6) has pitch that rises on the last word (more precisely, on the last stressed syllable) because it is a question. But a so-called "Wh-question"[4] has its stress on the "wh-" word (see Figure 5.7).

The sentences just used as examples are deceptively simple since it is easy to deduce correct intonation from the words and punctuation. Intonation is much

[3]The *schwa,* often represented as an upside-down "e" in dictionary pronunciation guides, is approximately the unstressed initial vowel in "abut" and is the "generic" vowel in English.

[4]"Wh-"questions begin with "Who," "What," "Where," "When," etc. They are in contrast to the questions that expect a yes-or-no answer.

Ex.1018 / Page 113 of 343
AMAZON.COM, INC.

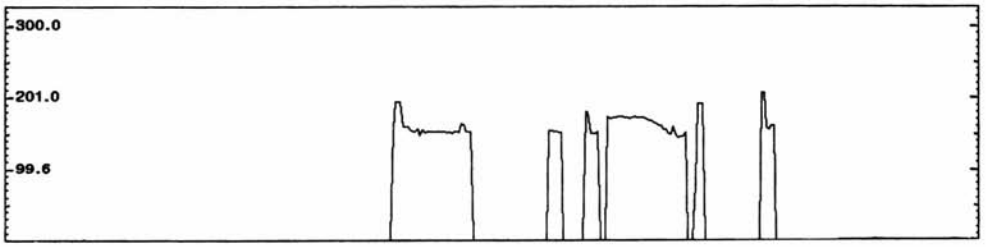


Figure 5.5. Pitch track of "*She went to Paris.*"

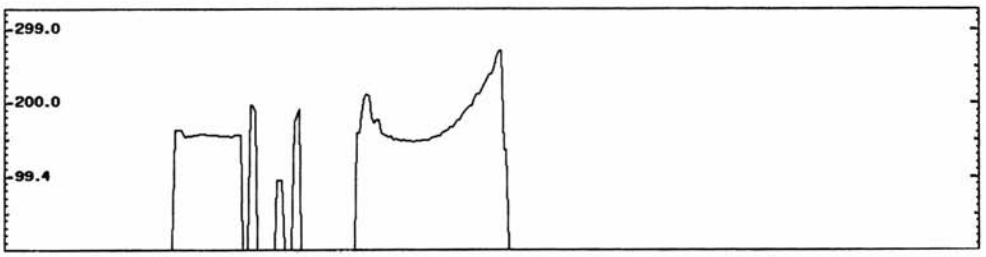


Figure 5.6. Pitch track of "*She went to Paris?*"

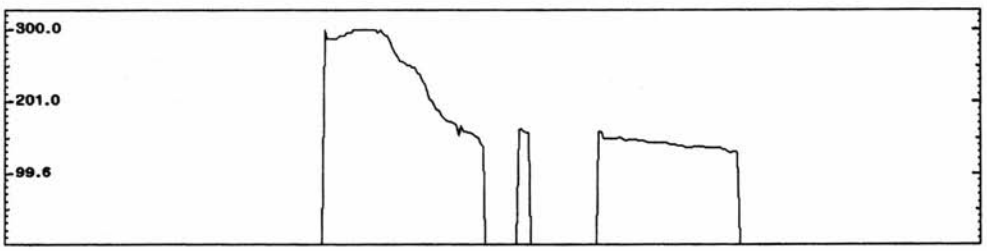


Figure 5.7. Pitch track of "*Who went to Paris?*"

harder to predict for longer sentences that are syntactically more complex; each phrase contributes its own intonation. Questions may be embedded in statements, and we cannot determine from orthography alone whether "*She went to Paris?*" is asking about the traveler (Who went to Paris?), the destination (Paris or Rome?), or verb tense (Has she gone there already?). In short, intonation is difficult.

Consequences for Synthesis

The factors just considered, namely lexical stress, coarticulation, and intonation reveal the limitations of the simplistic model of text-to-phoneme interpretation described earlier. To determine lexical stress, one must examine the underlying

Ex.1018 / Page 114 of 343
AMAZON.COM, INC.

morphological composition of the word. The realization of particular phonemes may change during the composition; for example, consider the "s" in "use" and "useful" as contrasted with "usability," in which the fricative is voiced. Further, identification of the stressed syllable is a prerequisite to application of coarticulation rules as unstressed syllables are more amenable to change.

Text-to-phoneme rules and lexical lookup generate a string of phonemes. Coarticulation may then substitute or remove phonemes or may call for allophonic variations; this suggests that allophones are a better intermediate representation than phonemes. Stress must then be realized as pitch and duration applied to these phonemes. Finally, sentence-level intonation adds an overall pitch contour above and beyond the syllable stress. To convey all this information, the string of phonemes or allophones must be marked with pitch and duration information.[5] Figure 5.8 summarizes the interaction of the components of more complete text-to-phoneme generation.

FROM PHONEMES TO SOUND

After text has been converted to a string of phonemes accompanied by prosodic information, the phonemes must be realized acoustically. Two approaches are used to generate the appropriate sounds to produce speech. The first method controls a digital vocal tract model by modifying internal parameters over time, while the second method pieces together small segments of digitized speech.

Parametric Synthesis

Parametric synthesis, which is also called **terminal analog** or **formant** synthesis, generates speech by varying the parameters that control a software vocal tract model; changing these parameters over time generates speech-like sounds. Vocal tract models for phoneme synthesis are similar to those outlined in the dis-

[5]This process is consistent with the discussion in Chapter 1 about how prosody does not fit cleanly into the layered model of speech communication.

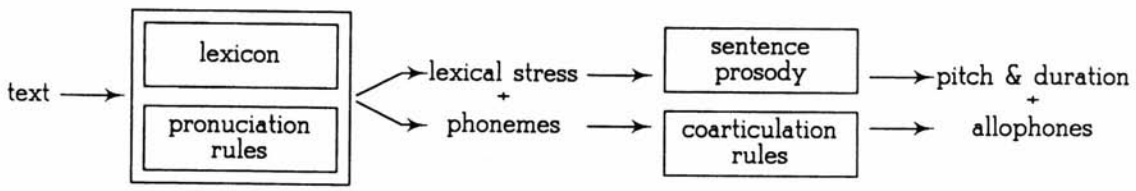


Figure 5.8. A refined diagram of text-to-phoneme reduction with output marked for pitch and duration.

Ex.1018 / Page 115 of 343
AMAZON.COM, INC.

cussion of Linear Predictive Coding in Chapter 3. This model includes a source and a number of filters to implement the transfer function (resonances) of the vocal tract. The source may be voiced (periodic) or unvoiced (noise); if voiced, the pitch (period) must be specified. This signal is then modified by a number of filters that produce the formants in the speech signal. Although the details of such a model are beyond the scope of this book (see the general references, especially [Klatt 1980, Klatt 1987a]), Figure 5.9 shows a partial list of the parameters used to control one of the early experimental synthesizers.

The vocal tract model produces a steady sound for a particular set of control parameters. It produces this sound by taking an input (a periodic or aperiodic pulse train) and filtering it to produce an output value; the parameters control the source and filtering functions. A new output value is produced 8000 or 10,000 times per second. The vocal tract model alone is not enough; its control parameters must be changed over time to mimic speech. The phoneme-to-sound portion of the synthesizer must model the dynamic character of phonemes by updating the parameters frequently (every 5 milliseconds is typical). The phoneme realization model takes account of how the vocal tract is configured to produce the spectra typical of the various speech sounds.

In many commercial synthesizers, the text-to-phoneme rules run in a general purpose microprocessor, and the vocal tract model runs as a continuous process on a digital signal processor. The control parameters are passed across a hardware interface between these two components. Considering the relatively small bandwidth of the control parameters, this is an effective task distribution in the hardware. However, general purpose processors have now achieved speeds where it is practical to implement the entire speech synthesis process on a workstation

```
Amplitude of voicing (dB)
Amplitude of frication (dB)
Amplitude of aspiration (dB)
Fundamental frequency of voicing (Hz)
First formant frequency (Hz)
First formant bandwidth (Hz)
First formant amplitude (dB)
Nasal zero frequency (Hz)
Nasal zero amplitude (Hz)
Nasal pole frequency (Hz)
Nasal pole amplitude (Hz)
Nasal formant amplitude (Hz)
Glottal resonator frequency (Hz)
Glottal resonator bandwidth (Hz)
```

Figure 5.9. Some of the control parameters for the synthesizer described in Klatt [1980]. These parameters control filters in the vocal tract model, e.g., the center frequencies and bandwidth of the various formants.

Ex.1018 / Page 116 of 343
AMAZON.COM, INC.

without any additional signal processor. This will lead soon to a new generation of all-software speech synthesis products.

Concatenative Synthesis

Consonants are generally more difficult to synthesize than vowels because of their short duration and dynamic behavior. Therefore, it should come as no surprise that the intelligibility of synthesized consonants, especially consonant clusters ("str," "sp," "bl," etc.), is lower than that of vowels [Spiegel 1988]. This problem has spurred an alternate form of phoneme realization called **concatenative synthesis** because it is accomplished by putting together short segments of recorded speech.

A concatenative synthesizer speaks by "gluing" together small units of digitized speech. What segments of speech should be used as the basic units? As stated at the beginning of this chapter, storage of each word in the language is impractical simply because there are too many of them. The next smallest unit is the syllable, but this too is impractical as there are over 10,000 syllables in English.

At the opposite extreme lies the phoneme. Any language uses a limited set of phonemes so storage would be minimal. But there are several problems with the use of phonemes as the base unit of concatenation. First, some of the voiceless consonants cannot be spoken independent of the associated vowel, rendering pure phoneme concatenation impossible. Second, it is exactly the dynamic nature of phonemes that motivates concatenative synthesis.

Since the dynamic and coarticulatory aspects of phonemes are most prevalent at their boundaries, this suggests a compromise consisting of a unit going from midway in one phoneme to midway into the next. This unit is called a **diphone** [Peterson *et al.* 1958]. Related units of speech for synthesis are the **dyad,** or vowel-consonant-vowel segments [Sivertsen 1961], and the **demi-syllable** [Fujimura and Lovins 1978], which is as the name implies half a syllable. Since there are about 40 phonemes in English, there are 1600 possible diphones as a diphone is simply a pair of phonemes. But not all these possible diphones exist in the language; for example, "sl" does occur in English, but "sr" does not. In practice, some extra diphones must be included to allow for allophonic variations and perhaps for differences between stressed and unstressed syllables.

In whatever form these units are encoded, it is important that their boundaries match closely. For example, if "bad" is to be synthesized by concatenating "ba" and "ad," the formants in the vowel portions of each must match or the discontinuity will be audible. Furthermore, it is necessary to be able to modify the pitch and duration of the stored diphones in order to support stress and prosody. LPC speech has often been used for concatenative synthesis because its pitch can be changed by modifying a single coder parameter, and segments can be lengthened by adding frames. Several successful diphone synthesizers have been built, perhaps the most notable (for English) by AT&T Bell Laboratories [Olive 1977]. At Bellcore (Bell Communications Research), Orator is a demi-syllable synthesizer that has been optimized to pronounce names [Spiegel et al. 1991, Spiegel and Macchi 1990].

Ex.1018 / Page 117 of 343
AMAZON.COM, INC.

QUALITY OF SYNTHETIC SPEECH

From the application developer's point of view, recorded and synthesized speech are very different because the latter allows an application to speak arbitrary text instead of a few prerecorded responses. However, from the end user's perspective, the two are functionally similar and share many problems relevant to user interface design; they will be treated jointly in the next chapter. In the user's experience there is one difference, however, that is difficult to disguise—intelligibility. Although some studies have compared synthesized speech favorably with very low bit-rate digital speech, few users are likely to encounter such degraded recorded speech except perhaps in children's toys. Although intelligibility is always a potential issue with speech output, it is especially pernicious with synthesized speech.

There are a number of possible sources of mispronunciation. The text-to-phoneme rules may be inadequate, or the morphological decomposition may break down resulting in an incorrect phoneme string. Lexical stress may be incorrectly assigned to confuse the noun and verb forms of a word or adjective and verb forms (e.g., "elaborate" and "live"). Coarticulation may be omitted or applied incorrectly, or prosody may be inappropriate. Incorrect intonation or duration rules can result in poorer listener comprehension than no prosody at all (monotone) [McPeters and Tharp 1984].

Even if the phoneme string is correct and marked with accurate pitch and duration, the phoneme-to-sound rules may not produce correct output. The consonant clusters are especially difficult to synthesize because of their dynamic aspects. Nasal consonants also pose problems because of the inability of simple vocal tract models to mimic the effect of the nasal cavity's absorption of some frequencies.[6]

Synthetic speech is sometimes likened to noisy recorded speech in terms of intelligibility. But as Nusbaum *et al.* discovered, the listeners make different mistakes when listening to synthetic speech [Nusbaum *et al.* 1984]. Different sets of phonemes are confused for synthetic or noisy human speech. Pisoni attributes the difference to the paucity of distinguishing cues in synthesized phonemes versus the rich set of cues in human speech [Pisoni *et al.* 1985].

Proper names are particularly difficult for text-to-phoneme analysis; this is due in large part to the broad ethnic diversity of names in English (at least in North America). Yet names are one of the more likely types of data found in text databases. Storing names in a lexicon is difficult simply because there are so many of them. Although a small number of common names covers a significant portion of the population, there are a tremendous number of names remaining; Spiegel [Spiegel 1985] estimates that although the top 5% of the names covers

[6]An all-pole model cannot cope with the zeros present when the nasal cavity is connected to the vocal tract.

Ex.1018 / Page 118 of 343
AMAZON.COM, INC.

90% of the population, there is probably a total of 1.5 million names in the United States. In addition, the names popular in one region of the country may be uncommon in another; German names are common in Milwaukee, Chinese and Hispanic names in Los Angeles, and Irish names in Boston.

How can rules be used to pronounce surnames? English pronunciation rules fail for many other languages, resulting in incorrect pronunciation of names derived from those languages. A better strategy is to identify the ethnic origin of a name based on spelling and then apply rules specific to that language. Even this may not be adequate as many foreign names have been Anglicized, and there are often several variants of the Anglicized form. Perhaps a suitable performance criterion is whether the synthesizer chooses a pronunciation of the name that may be plausible if spoken by a person. Vitale details a name pronouncing algorithm similar to that described here in [Vitale 1991].

Measuring Intelligibility

How can intelligibility be measured? There are a variety of methods, none of which reveals the entire picture (a good description plus some experimental results for all methods discussed here is found in [Pisoni *et al.* 1985]). A common test is the Modified Rhyme Test (MRT) [Fairbanks 1958], in which the listener is presented with a synthesized monosyllabic word (Consonant-Vowel-Consonant), and a set of six alternatives from which to choose the word that was spoken. Half of the stimuli vary the initial consonant (e.g., "hit," "wit," "mitt"), and the other half vary the final consonant. Thus, the MRT provides insight into the intelligibility of individual phonemes. With an "open form" of the same test, subjects are *not* given a list of choices, but instead simply indicate which word they heard. Not surprisingly, scores on the open response form are significantly lower than for the closed form. It should be noted, as pointed out by Spiegel [Spiegel 1988], that the MRT does not test intelligibility of consonant clusters, which we would expect would be particularly difficult to synthesize. More recent synthesizers score higher on the MRT than the older and less-developed ones, yet still do not approach natural speech.

However, we rarely listen to single words spoken in isolation; words are parts of a sentence or dialog, and we can often guess the next word even before it is spoken. By comparing listener transcription accuracy for semantically meaningful sentences (the Harvard psychoacoustic sentences [Egan 1948]) and nonsense sentences (e.g., the Haskins syntactic sentences [Nye and Gaitenby 1974]), we can see the contribution of meaning to intelligibility. As would be expected, semantically meaningful sentences are more intelligible than meaningless ones, but the difference is much greater for synthetic speech than for human speech [Pisoni *et al.* 1985]. This suggests that the semantic constraints on word possibilities play a stronger role in decoding the less intelligible synthetic speech.

The tests just mentioned measure the intelligibility of isolated words or single sentences based on the correctness and speed of a listener's response. For many applications, a synthesizer does not speak in isolation, but rather in the context

Ex.1018 / Page 119 of 343
AMAZON.COM, INC.

of a dialog or while reciting a longer passage of text. Listening comprehension tests have been used to quantify performance in such circumstances. Passages are either read by a human or synthesized, and subjects answer standard reading comprehension questions. Such tests have indicated that comprehension scores for synthesized passages may be as high or higher than for natural speech [Pisoni and Hunnicutt 1980], which suggests that general linguistic knowledge is a significant factor in comprehension.

Listener Satisfaction

Intelligibility metrics do not tell the whole story; other indications of listener satisfaction are more subjective and generally negative. Even if intelligible, synthetic speech is generally less pleasant to hear. This implies that it should be used sparingly, and in applications for which it will provide some clearly recognized added value to the user. Speech synthesis is still reserved for limited situations in which its poor sound quality is offset by the specific advantage of speaking from text.

It is important to note the difference between intelligibility and naturalness. Naturalness may be a particularly elusive goal, and, reassuringly, its achievement may not be required for many computer interaction tasks. On the other hand, unnatural speech may be tiring or boring to pay attention to. Poor prosody interferes with word intelligibility and, equally important, makes it difficult to attend to sentence breaks and the introduction of new material in a conversation. These are certainly important aspects of naturalness that impact listener satisfaction.

Performance Factors

Intelligibility and listener satisfaction cannot be summarized in single numbers. Although perhaps useful for comparison, ranking of synthesizers by intelligibility does not tell the entire story. Additional factors may contribute to the judgment on the viability of speech synthesis for a particular application.

One issue in user performance is the increased **cognitive load** associated with listening to synthetic speech. Because it is more difficult to decode the phonemes in synthetic speech, more demand is placed on the listener's short-term memory. If listening to synthetic speech takes short-term memory, this may interfere with whatever else the listener may be doing, such as keeping track of position in a menu hierarchy or remembering the information being spoken. In a series of experiments requiring subjects to memorize a string of numbers while listening to synthetic and then natural speech, Luce, Feustel, and Pisoni [Luce, Feustel and Pisoni 1983] found that subjects were able to recall the natural speech significantly better than the synthetic.

Another, possibly related, issue is **response time.** Because more processing is required to understand synthetic speech it takes longer. Measurements of subjects' response times in a task in which they were required to identify which word had been spoken showed significantly increased time for synthetic speech [Pisoni

Ex.1018 / Page 120 of 343
AMAZON.COM, INC.

1981]. This can also be detected using a method known as "gating," in which subjects are presented with short initial pieces of a previously recorded word (either natural or synthetic) and asked to identify it. A longer portion of a word is required for correct identification of synthetic versus natural speech.

On the positive side, there is strong evidence that understanding synthetic speech is a skill which can be learned and retained once learned. One experiment [Schwab *et al.* 1985] demonstrated that comprehension soared after a half-hour of daily exposure to synthesized speech over a 10 day period; this skill was retained 6 months later. If prospective users have sufficient reason to begin using synthetic speech, they quickly become proficient at its use.

APPLICATIONS OF SYNTHETIC SPEECH

For the reasons discussed in the previous section, synthetic speech is inferior to digitized natural speech as a computer output medium. Even when explicitly "mechanical" speech is desired,[7] digitized speech passed through various signal processing algorithms may be preferred to synthetic speech. For what purposes, then, is synthetic speech useful?

An obvious application is a reading machine for the blind. This concept started much of the original speech synthesis work in the 1970s and was the focus of several early commercial ventures. Such a reading machine relies on optical character recognition for input and synthetic speech for output. Even without a scanner, synthetic speech can be used by the blind and visually impaired to provide computer access. With their experience and increased dependence on the auditory senses, blind users learn to use synthesizers set at a very rapid speaking rate. Blind programmers are capable of writing and maintaining large programs with such an interface. Despite its importance and tremendous potential for the individuals involved, aid for the disabled is a rather small application niche.

Synthetic speech is valuable for prototyping an application using speech output, even though the final product will employ recorded speech. During interactive design, phrasings are liable to change significantly as the designer experiments not only with selecting the most appropriate words but also with how the system should articulate these words. Prototyping these systems with synthetic speech may save much time in the long run, as the speech output can be changed in the program source code with a text editor instead of recording a whole new set of prompts. Recording speech can be a difficult and time-consum-

[7]At the Atlanta airport, an automatic subway system moves passengers between terminals. When someone holds the door open, a voice announces that the train will depart the station when the doors are released. A monotone recorded voice was chosen for this recording to emphasize that no conductor was on board the train to intervene. The desired effect is that the passengers already on board the train glower at the offender and this intimidation generally frees the doors. From personal observation, the theory seems to be effective.

Ex.1018 / Page 121 of 343
AMAZON.COM, INC.

ing task typically requiring multiple attempts before resulting in a recording of agreeable quality.

For larger scale uses of synthetic speech, the indicators of a suitable application should be the size and complexity of the output vocabulary. The first class of appropriate applications for synthetic speech are those where the system has a relatively small repertoire of phrases it can say, but the words that make up these phrases come from a very large vocabulary. An example is an automated system to report an address for a given telephone number; the street name database could be huge. Automatic telephone directory assistance is a related problem due to the size of the name database and its diversity from city to city.

For applications such as these, the primary reason to choose synthetic over recorded speech is the logistics of building and maintaining the required database of voice recordings. Because the set of possible computer utterances is small, a number of whole phrases could be recorded with a few appropriate words substituted such as "The address is" "fifteen" "martin road." But it would be tedious to build the database, and both street and surname databases change over time. By the time the new recording is needed, the person who originally spoke the recordings may no longer be available, and switching voices in the middle of a sentence sounds confusing. With synthesized speech there is no need to record new words in the database, although a phonetic-exception dictionary may be required for those that are not pronounced correctly.

A second class of applications are those which generate natural language to describe some data in response to a query. Examples might include an electronic training manual describing how to diagnose and repair heavy machinery or a context-sensitive computer help system [Nakatani *et al.* 1986]. Such applications might be capable of generating a wide variety of utterances, and it would be difficult to exhaustively list and record them all. Synthesis of such material is sometimes referred to as **synthesis-from-concept** [Witten and Madams 1977].

The two types of applications just described are likely to be very different in terms of underlying software sophistication. The first is typified by simple queries into a very large database. The second type exhibits a much broader range of conversational ability arising in situations where the system is trying to convey in English some derived reasoning about the database. It is ironic that a sophisticated language-generation program may have a well-developed notion of syntax and the salience of particular words. This would be invaluable for synthesis of proper intonation but is lost when the utterance is reduced to a character string to be synthesized.

Finally, a third major area for use of synthesized speech is recitation of human-authored text. Since the text is created by a human, it is hardly predictable and cannot be limited to a small number of prerecorded words or phrases. One use of synthesis in this context is proofreading; it may be easier to hear errors than to see them especially for the author of a document. More widespread applications will come from telephone-based access to electronic mail or such common databases as a calendar or personal address book. Recently consumer products have appeared that use synthesis to speak traffic alerts, which are broadcast as text over pager frequencies.

Ex.1018 / Page 122 of 343
AMAZON.COM, INC.

SUMMARY

This chapter has served as an introduction to the technically difficult subject of synthesis of speech from text. The difficulty is due to the broad scope of the problem, ranging from linguistic analysis (parsing to find syntactic structure and prosody) to digital signal processing (vocal tract models and control parameters). It should be clear to the reader that speech synthesis is far from trivial and many factors influence the effectiveness of synthetic speech for the listener. There is much left to be learned about the fundamentals of speech production and the acoustic characteristics of natural speech. Speech synthesis will continue to improve, although progress may be slow and take place across a number of research areas.

There are some applications for which speech synthesis is superior to recorded speech, or where the latter simply is not adequate. Because of the steep learning curve encountered with synthetic speech, the prospective user must be enticed into initial use. The initial negative reaction to the speech quality can be overcome by an appropriate application with a well-designed user interface. The next chapter discusses in detail some guidelines for the development of such applications and offer examples of working systems as a means of exploring these issues.

FURTHER READING

Much of the pioneering work in speech synthesis in the United States was done by Dennis Klatt and Jonathan Allen at MIT's Research Laboratory in Electronics. In Allen *et al.* they describe along with coauthors the algorithms of MITalk, one of the early complete synthesis schemes. Klatt provides an excellent although slightly dated survey of international speech synthesis research in Klatt [1987b]. A section of Furui and Sondhi 1992 is devoted to speech synthesis with an especially strong overview by Allen. Witten [1982] provides a valuable perspective on speech synthesis in Witten, which is also a bit dated.

Ex.1018 / Page 123 of 343
AMAZON.COM, INC.

6

Interactive Voice Response

This chapter is about building *interactive* computer systems that use speech, either recorded or synthesized, as their sole form of output. The focus will be on the advantages and liabilities of speech as an output medium and on interaction techniques for applications to employ speech output most effectively. It is essential to understand the limitations of voice output, both when deciding whether speech output is appropriate to an application as well as when designing details of the interface.

Chapter 3 discussed techniques for digitally encoding and compressing stored voice, and Chapter 5 described the process of converting text to speech. This chapter considers attributes of both digitized and synthesized speech as components of interactive systems. Although most of the growing number of commercial voice response systems employ recorded speech, this chapter also includes applications of synthesized speech, which are emphasized in the case studies.

Chapter 4 discussed the role of stored voice as a data type, primarily in the context of its role as a compbnent of an electronic document. For most of the examples in that chapter, a visual user interface was employed because most documents contain visual material. In this chapter we consider applications with no visual interface by focusing on telephone-based interactive voice response systems. Such systems are interactive because the caller employs touch tones, or occasionally speech recognition, to make selections, and the application responds appropriately using speech output.

For the purposes of this chapter it is assumed that user input is accomplished with touch tones as the basics of speech recognition are not discussed until Chapter 7. Touch tones provide an abstract user input, that has no innate meaning

Ex.1018 / Page 124 of 343
AMAZON.COM, INC.

except in the context of an application. In contrast to speech recognition, touch tones are unlikely to be detected incorrectly by application hardware; this allows some simplification in discussions of user interaction techniques. This chapter ignores the technical details of the operation of telephones and transmission of touch tones except as they impact interaction techniques; details of the operation of telephones and telephone networks is included in Chapter 10.

This chapter also contrasts the assets and liabilities of speech as an output medium. After this, the focus shifts to details of interaction techniques for voice output and then to issues in menu selection or data entry for telephone-based voice systems. The final section offers three case studies as sample applications illustrating the techniques described in this chapter.

LIMITATIONS OF SPEECH OUTPUT

Chapter 3 raised the issue of intelligibility of recorded speech, mostly as an issue of speech coder design or selection. In the discussion of speech synthesis in Chapter 5, intelligibility was again highlighted as a major item of concern. Because synthetic speech is generally of poorer quality than encoded human speech, sources of pronunciation error and effects of listener learning are important when using speech synthesis.

Although intelligibility is certainly one factor in determining the appropriateness of using voice output in a particular environment, there are a number of other issues that also need to be considered in designing a user interface that uses any form of speech output. Speech is both slow and serial as well as "bulky," requiring listener attention to scan the contents of recorded voice files. Speech broadcasts within an acoustic environment, which raises issues of personal privacy and possibly disturbs one's neighbors. These characteristics are not limited to synthetic speech although they may be exacerbated by its decreased intelligibility and the suspected increase in cognitive load on the listener. These issues, as much as intelligibility, place constraints on whatever interaction techniques might be employed by an application.

Speed

Speech is slow; listening is much slower than reading. Typical speaking rates are in the range of 175 to 225 words per minute. People can easily read 350 to 500 words per minute. A 9600 baud computer terminal can receive nearly 10,000 words per minute, while a 1200 baud modem (nearly obsolete in current communications technology) transmits more than 1000 words per minute. Both recorded and synthesized speech can be sped up, but time-scaling to more than 1.5 to 2.0 times the original speed reaches an upper limit and requires considerable attention by the listener. Reading rates can be increased with practice as well.

As a consequence of its slow speed, using voice to access information can become quite tedious. A few paragraphs of text may take a minute or more to speak. The information to be spoken, either by synthesis or digital playback,

Ex.1018 / Page 125 of 343
AMAZON.COM, INC.

needs to be concise and direct. One of the banes of voice mail systems are loquacious users who leave long, rambling messages. Because of speed considerations some text-based interactive computer activities such as browsing through network news may be unsuitable for speech output.

In comparing the speed of speech to conventional screen and keyboard interaction, it is important to note that the speed difference is not symmetrical with respect to input and output. Although listening is slower than reading, speaking is faster than writing or typing in many situations. For example, the author of a recorded voice message has a speed advantage over the listener; this may likely influence the circumstances under which voice may be chosen over text if both media are available.

Temporal Nature

Speech is an acoustic phenomenon consisting of variations in air pressure over time. By definition, once spoken the speech is gone. If the listener was inattentive or absent when a message was spoken the information is lost. This is the antithesis of graphical user interfaces with dialog boxes that persist until the user clicks on a mouse button. The temporal nature of speech requires a means of focusing the listener's attention to attend to sporadic auditory events; many international airports play a distinctive chime before speaking a boarding announcement over the public address system.

Serial Nature

A stream of voice conveys only one word at a time;[1] we must wait for the first word to be spoken before we can hear the second. If a menu is displayed on a terminal, all the items appear very quickly and the user then visually scans them until a choice is made. With speech each item must be recited in turn; once spoken it is gone unless the user remembers it. If the user misses a portion of the message, it must be repeated in its entirety. Because this serial nature of speech places additional memory requirements on the user, menu items and other application responses must be short and precise.

A consequence of the temporal nature of speech and the serial presentation of menu items is the interaction between presentation sequence and the timing of user input. As shown in Figure 6.1, it may be difficult to detect which choice a user responds to if expected to make a selection immediately after hearing the desired choice.

Bulkiness

The storage space required for recorded voice was of much greater concern five or ten years ago than today. As the costs of computer memory and magnetic disks

[1]This is not quite true. Speech is not just a string of disjoint phonemes; there is a certain amount of parallel transmission of phonemes in that the allophonic and acoustic realization of a phoneme is influenced by its neighbors.

Ex.1018 / Page 126 of 343
AMAZON.COM, INC.

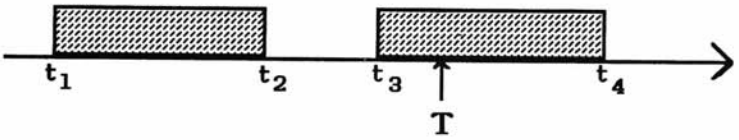


Figure 6.1. Difficulties synchronizing user input with speech output. Two utterances are spoken spanning t_1 to t_2 and t_3 to t_4. At time T the user responds. Is this response a delayed reaction to the first utterance or a very quick response to interrupt the second?

have dropped, storage has become a less important issue. However, managing stored voice and providing the user with sufficient access to it remains a difficult problem for other reasons.

If we wish to hear information recorded in a sound file, how can we determine which file to play, or what part of the file to play if it is many seconds or minutes long? Retrieval can be facilitated if a text-based description accompanies the sound; this can be stored as "header" information in the sound file itself. Situational cues, such as when the recording was made, the identity of the speaker, and which computer application created the recording may also help limit the number of files that must be searched. But such information is of marginal additional value and in no way approaches the power of keyword searching among text files.

Finding a small number of words in a large quantity of speech is known as keyword spotting; unfortunately, this is one of the most difficult forms of speech recognition and currently not widely available. Instead, a person must listen to the recorded speech to find a particular piece of information. Listening requires *attention*. Although the user can perform other tasks while executing a long text-based keyword search, this is not possible while listening.[2] Although both synthesized and digitized speech may be sped up during presentation, this demands still more attention from the listener.

Privacy

Voice announcements, messages, or warnings are much more public than visual notification. A message displayed in a small font on a computer screen may be visible only to the typist or someone sitting nearby, but voice messages may be heard across a room. In addition, if a visitor to an office tries to read the screen, this is apparent to others by their gaze; eyes are more directional than ears. Excessive use of voice output can lead to a distracting and tiring work environment, and sensitive messages should never be voiced without a guarantee of privacy. But this characteristic of speech can also be advantageous in some classes of applications; speech is useful in public address systems precisely because

[2]We *can* perform other visual or manual tasks while listening but still must devote significant attention to the auditory channel.

Ex.1018 / Page 127 of 343
AMAZON.COM, INC.

everyone can hear the announcements regardless of the focus of their visual attention. Voice can be similarly used to alert the computer user in an office when the user requires mobility or is not paying attention to the screen.

ADVANTAGES OF VOICE

Given these significant liabilities of voice, what advantages could possibly justify its use? Chapter 4 suggested several roles of voice in conjunction with text in the context of document-oriented applications. Voice is richer than text and more expressive of information that may be tentative or subject to multiple interpretations. Voice is useful for annotation because it acts as an added dimension for placing comments on an otherwise two-dimensional view of a document page. The rest of this section describes assets of voice beyond document-oriented uses.

For many styles of application discussed in this chapter, the main advantage of speech is its **accessibility** via telephone-based interfaces. Telephone-based applications can be used from nearly any telephone. Most public telephones in the U.S. and Canada are equipped with touch tone keypads, but this is not true in some other parts of the world, notably some European countries. When touch tones are not available, speech recognition must be employed; unfortunately, speech recognition over the telephone is much more error prone than touch tone decoding.

The accessibility advantage of voice also applies to the authoring or creation of information. The ability to capture speech as a recording for subsequent playback allows data to be entered into a system when no other means of input is available, such as recording notes on a pocket tape recorder, dictation, or leaving messages to oneself on one's answering machine. Telephones allow callers to record information at a distance; for example, Resnick describes a community bulletin board enabling callers to browse as well as enter events in a public events database by recording them [Resnick 1992a]. Voice mail allows subscribers to record messages, and Chapter 12 describes telephone access to desktop computer applications that allow a user to record voice entries in a personal calendar. Telephone-based recording also can be used for functions such as order entry by filling out an audio "form" for later transcription.

Although the previous section describes the **broadcast nature** of speech as a liability, in other circumstances it is an asset. The ability to reach a number of listeners simultaneously over a broad area, such as airport flight departure announcements, simply could not be provided by visual displays.[3] Because our ears are much less directional than our eyes, auditory cues can be delivered with more confidence that the intended audience will be more able to be attentive at the moment of presentation. Voice output is effective when the user is mobile or

[3]It should be noted that displays clearly are of value. Imagine having to locate your departure gate by listening to a continuous recitation of all flights departing within the next hour!

Ex.1018 / Page 128 of 343
AMAZON.COM, INC.

less likely to be focusing on a display. The user may be staring out the window, reading a book, or walking around the office and not notice a new message on the computer screen.

Voice can also be useful even when the intended recipient is in a more restrained physical environment such as sitting in an office using a workstation or in an airplane cockpit, because it provides an alternate channel of interaction when the user's **hands and eyes are busy** elsewhere. If appropriate and informative words are spoken such as "System shutdown in ten minutes" or "New mail from Kaya," then users would not need to shift their gaze to the screen.[4]

Spoken messages offer a performance improvement in addition to providing the ability to audibly distinguish the message content. Experimental evidence (see [Allport *et al.* 1972, Wickens 1981, Treisman and Davies 1973]) indicates that if a user is performing multiple tasks, performance improves if those tasks can be managed over independent input/output channels, or modalities. For example, although we cannot listen to two conversations simultaneously, subjects were able to sight read piano music while repeating what they heard with no decrease in proficiency [Allport *et al.* 1972]. How do such experimental results apply to voice in an office environment? Voice notification to indicate the status of a task to which the user is not paying immediate attention or for which the associated window is not visible could effectively deliver a message with minimal disruption of the visual attention a user is simultaneously applying to other tasks.

Speech synthesis affords systems the capability of **media translation** to turn text documents into voice. Although this seems to be a trivial continuation of the earlier discussion about remote access, it should not be dismissed quite so quickly. Much of the current work in multimedia computing uses different media for distinct purposes to present disjoint, although related, information. For example, a voice narration accompanying the display of a simulation program is meant to comment on the program's graphics rather than act as a substitute output medium. Speech synthesis lets text data be converted into voice for any purpose. Some blind users make remarkable use of it. Voice output is a first step towards employing multimedia technologies to make computers accessible to a variety of users under many different circumstances.

DESIGN CONSIDERATIONS

An understanding of the limitations of speech as an output channel enables us to design better voice interfaces. Speech can be used effectively for computer output, but care must be taken to avoid allowing the liabilities of the medium to overcome the usefulness of the application. The next two sections of this chapter

[4]Displaying the text message *in addition* to speaking it is necessary for several of these applications. If I am out of the office when an important notice arrives, I will miss the speech output but may look at the screen when I return.

Ex.1018 / Page 129 of 343
AMAZON.COM, INC.

outline a number of design considerations and discuss some interactive techniques in this light.

Most of the limitations of voice output described earlier can be summarized by the broader observation that speech takes place *over a period of time.* The amount of time required to listen to a voice recording, the necessity of reciting menu items in sequence, and much of the difficulty associated with the bulky nature of voice are all a direct consequence of the temporal nature of speech.

Speed is thus one of the primary factors distinguishing voice from other interface media; for example, most graphics applications can be written with the assumption that display operations such as drawing lines or text are performed nearly instantaneously. Much of the time the application is idle while the user reads text, picks a menu item, etc.; this is punctuated by sporadic user input and display update. For telephone-based interfaces, the exact opposite is true; presentation accounts for most of the total time the user interacts with the application.

Voice response systems must therefore strive to minimize the amount of time required to retrieve information: they must be brief, selective about content, and interruptible. Applications must plan what to say so as to avoid utterances that are irrelevant by the time they are spoken. A real-time speech output system may have to formulate conflicting goals and choose among them if there is not sufficient time to say everything, which requires estimating how long it will take to speak an utterance. The awareness that speech takes time (and acceptance that time is a commodity of which we never have enough) should permeate the design process for building any speech application. This argues in favor of shortening spoken menus and precludes many applications that may be quite feasible in text such as browsing through a newspaper. Time motivates strategies to filter an application's output requiring careful judgment as to the relevance of a specific system comment. The remainder of this section explores these design considerations in more detail.

Application Appropriateness

Because speech is so difficult to employ effectively, discretion is essential in determining effective voice applications and in choosing the appropriate speech output device. Poor quality speech, whether synthesized or recorded, quickly becomes annoying unless it serves a clear purpose. For example, speech output has been employed effectively as a warning medium in airplane cockpits. On the contrary, few car drivers appreciate a tinny voice reminding them to close the doors of their automobile. The cockpit is a busier place, and the consequences of an error may be more severe. A more effective use of speech in the car may be traffic information (drivers already tune to commuter traffic reports on their radios) or as part of a navigation system (as described later in this chapter).

In retrospect, another example of a misguided speech interface is the recitation of a customer's purchases at a cash register. As each item is scanned by the bar code reader, its description and price are announced. The original motivation for

Ex.1018 / Page 130 of 343
AMAZON.COM, INC.

these systems was a concern over customer skepticism at the accuracy of the bar code readers. However, public acceptance of this technology came rapidly, and as a result the voice output soon was considered annoying, both to the cashier who had to listen to this drivel all day as well as to the customer who perhaps was not interested in having his or her purchases announced to the surrounding patrons. A text version of this information, both on a register display as well as on a detailed receipt, adequately satisfies customer doubts.

Speech offers more opportunities in telephone-based interfaces because it offers a means of remote access to important information. Such systems are the focus of much of the remainder of this chapter. However, for such applications to be useful, access to the information being requested over the phone must be a priority to the user. Under most circumstances, few sighted computer users would choose to hear their electronic mail synthesized when they could read it. But timely access to certain messages may be important enough to recipients to make telephone access to messages a valuable application.

Speech quality interacts with the perceived utility of an application. If both can be used equally effectively, it is rarely the case that synthetic speech is more appropriate to an application than recorded speech. If only a few spoken messages are needed, prerecorded speech generally is adequate, less expensive, and likely to be better understood. For some data, however, synthesis is essential.

Data Appropriateness

Since speech is slow, only essential data should be spoken. For example, while speaking electronic mail, such information as dates, times, and many other extraneous fields detailing message routing should be discarded. The point is less "Is this data useful or interesting?" than "Is this data *really* necessary?" Removing extraneous information can save the listener a substantial amount of time and increase the likelihood of acceptance of the application. Information that is spoken should be presented in a simple and accommodating form. Computer time stamps often include the date and time in seconds or even milliseconds, but a person likely prefers to hear "this morning at 10:30" or "yesterday evening" instead of "Wednesday October two 1991 at seven twenty-one and twenty-three seconds."

The wording of recorded speech output is also critical. Prompts should be as brief as is practical; excessively friendly or chatty systems wear thin. Some voice mail systems offer their users the option to choose between normal and abbreviated prompts that allow experienced users a more streamlined interaction. Whether a system is oriented towards novice or experienced users should depend on the expected user population. A service such as national weather information may have a broad base of occasional users, but a service to report the availability of a particular computer system may experience frequent use by a small number of individuals who benefit from terse phrasing.

If the data can be spoken as many variations in a sequence of information elements such as the set of 10 digits spoken as components of a telephone num-

Ex.1018 / Page 131 of 343
AMAZON.COM, INC.

ber, recorded speech is superior to synthesized voice. An example of such an application is telephone presentation of weather information; a forecast could be composed from small units such as "The weather in . . . Boston . . . is partly cloudy . . . with a temperature of . . . seventy-eight degrees."

Because listener comprehension of synthetic speech improves rapidly with exposure, frequent users of specialized applications may find it acceptable regardless of the data. If the data is human-authored, e.g., electronic mail or entries in one's calendar, synthetic speech is the only practical means of speaking. The data may be difficult to synthesize well, however. Surnames are particularly hard to pronounce correctly as they are derived from many languages. Complete sentences such as those found in electronic mail are more easily comprehended than isolated words or phrases that might be found in a meeting agenda announcement.

For all these reasons, the data must drive presentation strategies. Filtering, terseness, and use of an appropriate form of voice output are all essential.

Responsiveness

User input should trigger some voice response, either an acknowledgment of the input or actually speaking the information the user just requested. But because speech is slow, explicitly echoing every keystroke is not appropriate. If an application offers frequent feedback, then the user can be confident that input was detected; application silence might cause the user to press a key repeatedly and lead to errors.

If a system employs multiple menus, feedback can be a form of navigational cue by which each menu identifies itself ("top level," "main menu," "configuration menu") before presenting its choices. Such a cue would also be appropriate if the user enters ill-formed input; the system can remind the user what it thinks he or she is supposed to be doing. ("You are entering a telephone number. Please enter the number terminated by a pound sign.") Similar vocal prods can also be made during the invocation of time-outs such as when the user does not make a selection after hearing an entire menu of choices.

Speech Rate

One method of increasing an application's apparent responsiveness and minimizing message retrieval time is to speed up voice output. Most speech synthesizers support commands to vary speech rate, expressed as words per minute. Informal observations of experienced users accessing electronic mail over the telephone at the Media Lab indicate acceptable comprehension at 300 to 350 words per minute with a synthesizer with a default speech rate of 180 words per minute. Blind computer users can understand significantly faster synthesized speech. Recorded speech can be sped up for playback as well as described in Chapter 3.

Time compression of speech output is most effective for experienced users who are already familiar with the application, especially its menus and interaction structure. The more predictable the output, the faster it can be spoken.

Ex.1018 / Page 132 of 343
AMAZON.COM, INC.

Interruption

A voice interface should almost always be interruptible. This is a direct consequence of the slow transmission time for a message. A passage may be irrelevant, unintelligible or redundant, and the user should be able to skip over it. If the system can detect input while speaking by allowing "type ahead" control (most commonly controlled by touch tones), an experienced user need not endure a long-winded introduction or menu repetition. This allows a clean compromise between abbreviated instructions for expert users and lengthier tutorials for novices. Unfortunately, users tend to be more hesitant to interrupt than system designers might expect or desire. It may be that users lack confidence in their ability to recollect the proper keystrokes or simply cannot be bothered to learn them, so it is helpful to keep explanations short.

One problem with supporting interruption is that it may be difficult for the system to determine exactly where the interruption is intended; this was alluded to earlier in this chapter. There is a time delay of at least several hundred milliseconds between the transmission of text to the synthesizer and the onset of speech output, and an additional delay while the user makes a decision; this is a challenge when building an interface that presents choices as "When you hear the item in which you are interested, press any key." Even the slightest of delays due to the user's processing of the information may result in an erroneous interpretation as the next prompt begins to play.

Another problem with interruption is that it may not be possible in a telephone-based system utilizing voice recognition for user input. Analog telephones are two-wire systems; the voice from each party is mixed in the circuit. This implies that the speech recognizer hears the system's transmitted speech as well the user's voice, and this prevents reliable recognition because the recognizer is not able to discriminate between speech sources. Touch tone recognition is not as difficult as it is acoustically dissimilar from speech.

A solution to this mixed channel problem is available to allow the user to "talk over" the system's speech output to the speech recognizer. Since we know which signal corresponds to the spoken speech output, we can "look through" the signal by using it as a reference and sampling the telephone line during short silent periods in the system's speech. These silent periods may correspond to some of the longer stop consonants or pauses that occur between phrases. If a significant signal is detected by the computer at these times, it must be the user trying to interrupt; speech output can be stopped so recognition processing can be begun. Such a configuration, shown in Figure 6.2, was built and proved successful in limited testing in the author's laboratory.

Repetition

A means must be provided for the system to repeat itself. This applies both to reciting menus several times as well as to repeating sentences of human-authored text or computer-generated discourse. Repetition will likely be combined with an interruption strategy if long messages are being recited.

Ex.1018 / Page 133 of 343
AMAZON.COM, INC.

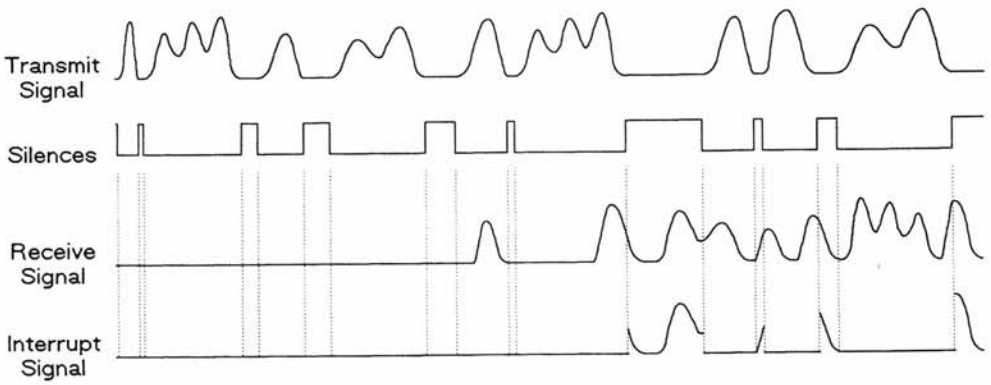


Figure 6.2. A configuration to "look through" transmitted speech to detect user interruptions.

How should a repetition be spoken? In the case of a menu, it suffices to repeat the menu items with identical wording as the items are often short and the listener most likely was just having trouble recalling them all. If output consists of recorded speech, the application can only replay the recording, possibly doing so at a slower speed. If the text is a passage of computer-generated discourse, it may be effective to generate an alternate phrasing of the text in the hopes that this would help with any pronunciation errors. However, this is also problematic as the user may be expecting a word-for-word repetition and thus be put off balance. There is little empirical evidence one way or the other.

In the case of synthesizing human-authored text, however, the problem could be either a spelling or typographical error, failure of the synthesizer's text-to-phoneme rules, or lack of user attention. One repetition strategy is to speak the same text again at a slower rate in the hope that this enhances intelligibility. The system to read electronic mail over the telephone, which will be described as a case study, chose to repeat more slowly and then to switch to "spell mode" and pronounce the text letter by letter. Spelling a mail message or almost any other data is tedious. An alternative strategy involved passing the text passage through a spelling checker program and during repetition combining a slower speech rate with letter-by-letter spelling of any words not found in the dictionary. This technique helps catch text-to-phoneme errors as proper names are likely to be pronounced poorly but would not be found in the dictionary. Most typographical errors would also fail the spelling check and thus would be spelled out.

Exception Pronunciation

Words may be pronounced incorrectly. Ideally, the programmer should be able to add to the morpheme dictionary in the synthesizer, but no synthesizer product currently provides this capability. Some synthesizers allow for alternative pronunciations of a small set of words to be downloaded into the synthesizer, but if the synthesizer only uses this list with simple string substitution, it is not a

Ex.1018 / Page 134 of 343
AMAZON.COM, INC.

very powerful capability. For example, if "Schmandt" is in the dictionary, will "Schmandt's" be pronounced correctly?

Because of their limited capacity and literal string translation (instead of direct application access to the morpheme pronunciation tables), synthesizer-based exception dictionaries are not very powerful. Improved functionality can be provided in more capable workstations that might employ speech synthesizers. In fact, many workstations are now able to perform real-time speech synthesis entirely with software.

Instead of using simple exception lexicons, applications can better benefit from domain-specific text-to-phoneme rules. Consider several examples. My phone number, spelled 253-5156, should not be pronounced "two hundred fifty three hyphen five thousand one hundred fifty six." If the string "253-5156" is known to be a telephone number, it can be pronounced properly as three digits, a pause, and four digits. Postal Zip Codes are similarly pronounced digit by digit but street addresses are pronounced as ordinary numbers. My email host, "media-lab.mit.edu," is pronounced "media hyphen lab dot m i t dot e d u" by those familiar with Unix and Internet addressing.

While it is difficult to employ this knowledge to speak free-form human-authored text, much of the data to be synthesized is highly structured. Electronic mail headers, name and address databases, telephone lists, and appointment calendars each contain much structured data, which can benefit highly from string processing before synthesis. In this way, applications can employ much more intelligible synthetic speech.

Multiple Voices

Many synthesizers have the ability to use one of several vocal tract models or "voices." Switching voices may be invoked by software that offers a selection between male, female, deep, or raspy speech. The use of several different voices *may* make it easier for the user to discern various separate system functions, such as a command from data or a narrative mode from a multiple-choice mode. Realistically, however, even though a synthesizer may support multiple voices, a primary voice generally emerges as being significantly more intelligible than the auxiliary voices. Rosson studied users' responses to synthetic voices while varying a number of characteristics of the vocal tract model; she discovered that users preferred different speech qualities for different tasks, which she categorized as information providing, entertainment, and feedback [Rosson and Cecala 1986].

Little work has been done to determine how effective such a technique is in practice although frequent users of synthetic speech often modify pronunciation. For example, a telephone pager at the Media Lab uses speech synthesis to announce calls. Each user of the system is able to configure his or her own personal announcement message, and some users have included in their message the necessary escape sequences to change the synthesizer's voice to better differentiate their call announcement from others. Other users achieve a similar effect by modifying the prosody of their message or the duration of some of its syllables.

Ex.1018 / Page 135 of 343
AMAZON.COM, INC.

USER INPUT WITH TOUCHTONES

This chapter emphasizes telephone-based applications of speech output. These applications use touch tones for user input; when the user presses a button on the telephone keypad, a tone is generated that is decoded into one of 12 symbols at the application end, usually using special but inexpensive hardware. This section discusses a variety of user interaction techniques employing touch tone input.

User input can be divided into two classes: making a selection and entering data. Menus are a selection mechanism, and the response to a yes-or-no question can be thought of as a menu limited to two choices. Data entry may be numeric using the obvious numbers on the telephone keys or may involve the spelling of words by making use of the letters that also appear on most keys. As already noted, telephones outside of North America may not have letters in addition to numbers.

When presenting menus, how should keys be described? Although most references are straightforward, a few are confusing. The "0" key is likely to be called "oh," both because that is how the digit is usually pronounced in phone numbers as well as because it is used to reach the operator. The "#" key is alternately called "pound" (as in weight), "sharp" (from music notation), or "number sign"; in the telecommunications industry it is very rarely referred to as the "octothorpe." The key labeled "*" may be called "star" or "asterisk."

Menus

Menus may be either **temporal** or, more commonly, **enumerated.** In an enumerated menu each choice is presented accompanied by a number: "Press one to hear your messages, press two to send a message, press star if you are finished." A temporal menu asks the user to press a key when he or she hears the choice desired and pauses between each choice: "Hit any key when you hear the city you want. Somerville . . . Cambridge . . . Boston. . . ." A binary menu (i.e., yes or no) is simply a degenerate case and may be implemented in either way. If a default is to be assumed, then a short form can be used such as "If you want to send a reply, press any key now."

Each style has both advantages and disadvantages. Timing is more critical with the temporal menu style because the application must know when the prompt was spoken and measure a consistent amount of time before presenting the next choice. In the event of a user entering input just at the beginning of the speaking of a choice, it may be unclear as to whether the user's response was an early acceptance in anticipation of the next choice or a delayed response to the previous choice. The enumerated menu also needs a timer to handle circumstances in which the user makes no response to the enumerated list; the usual approach is to repeat the menu after the predetermined time-out period elapses.

An advantage of the enumerated over the temporal menu is that the experienced user need not listen to the entire list but can interrupt its recitation by entering a choice. The temporal menu necessitates waiting until the desired

Ex.1018 / Page 136 of 343
AMAZON.COM, INC.

choice is presented.[5] But if the list of choices changes (perhaps it is a list of voice mail users and a new employee joins the organization), then the associated numbering may change and the experienced user who types ahead may make an incorrect selection. The enumerated menu also requires the user to keep track of both the choice and the number associated with the choice; with a temporal menu one can concentrate exclusively on the list items themselves.

An issue with all menus is their size. Human factors folklore suggests that four items is the comfortable number. In a system implementing voice mail for athletes at the 1984 Olympics, Gould *et al.* found four items to be too many based on evaluations of a prototype system [Gould *et al.* 1987]. In a somewhat more contrived test task based on a laboratory experimental setup, Englebeck and Roberts [Englebeck and Roberts 1989] confirmed the proposal of four items, but note that a larger number of items may be more effective for experienced users. In their study, two important factors influenced the choice of an acceptable menu size: interruptions and user confidence. Although users are normally hesitant to interrupt the system, as menu items increase users appear more likely to interrupt when their choice is heard. This interruption generally results in faster selection time, although it could indicate frustration at the long menu. Confidence in the choice about to be made is also key and suggests that the phrasing of menus is very important. When the user hears a choice that is clearly the desired selection, interruption is more likely.

With an enumerated menu another issue is whether it should be presented in the order of action . . . key ("Reply press one") or key . . . action ("Press one to reply"). According to the Englebeck and Roberts study, users were more likely to interrupt with the action/key presentation. Additionally, this order was preferred in the subjective evaluations of tests using the two options.

Data Entry

Under some circumstances user input is required and a list of menu choices is impractical simply because there are too many possible selections (such as looking a name up in a company telephone book). In this case, to gather user input the system must speak a prompt, receive the responding keypresses, and detect the termination of the input. User input may be **numeric** or **alphabetic.**

Numeric input has many uses and is relatively easy to explain to the user as we are all quite used to dialing numbers to place calls. As part of an interactive service, a user may be required to enter a credit card number, telephone number, or the digits of an address. Services offering weather reports for locations around the country may ask for the telephone area code for the location. Interactive banking services prompt for an account number or a dollar amount.

Alphabetic input is somewhat more awkward. Fortunately, in North America the telephone keypad has letters inscribed on the keys remaining from the days

[5]A temporal menu can eliminate the wait by allowing the user to press a special key to skip to the next item [Resnick and Virzi 1992].

Ex.1018 / Page 137 of 343
AMAZON.COM, INC.

when telephone exchanges were referred to by a name instead of number.[6] This alphabetic labeling tends to get carried over into other public keypad style devices such as automated bank teller machines. But there are two problems with this arrangement: each key maps into three letters (see Figure 6.3), and the letters "Q" and "Z" do not appear on any key of the telephone.

For the two missing letters several options are in use. Both letters may be mapped to the otherwise vacant 1 key; this seems the most common alternative. They are also sometimes mapped to the 0 key. For the implicit key-to-letter decoding, described below, they can be assumed to occupy the keys on which they would appear if they had been left in the alphabet. In this scheme, Q appears invisibly on the 7 (PRS) key and similarly Z appears on the 9 key.

Alphabetic input techniques may decode the three-to-one letter-to-key mapping either explicitly or implicitly. Explicit mapping requires the user to type two keys for each letter; the first key chooses a group of three letters and the second selects one from this set. The second key may be constrained to one of the set 1, 2, or 3 or better could be any key with the column position of the key selecting the first, second, or third letter. For example, the letter H might be selected by entering 4 5; 4 selects the GHI set, and 5 from the middle column selects the middle letter from the triplet.

The implicit letter mapping technique accepts the confusability of the user's input and attempts to decode the input by matching it against the set of valid inputs. For example, while selecting a name from a menu (see Figure 6.4), the user is really picking one of a set of names, each of which may have a distinct touch tone "spelling" uniquely specifying that each letter is not necessary. In fact, in the illustrated example, four of the names would be uniquely specified after the first touch tone. (One might, however, wish to gather all five digits to confirm that the user was not trying to select a name not on the list.)

Explicit spelling requires twice as many keystrokes but is required for some tasks, e.g., if a new subscriber to a service is asked to spell his or her last name to create a new account. In the case of spelling to make a selection from a list, implicit character mapping may be quite adequate, faster, and easier to explain to the user; examples include looking up a last name in an electronic database, spelling a street name, or logging in to a computer account.

A problem with implicit spelling is the degree of overlap between the items being selected and their keypad equivalent. For example, the login names "Leo" and "Ken" are both spelled "536." In such a case, the collision must be detected by the system and an alternate selection mechanism invoked, e.g., "If you mean Leo, press one; if you mean Ken, press two." (Note that this is a menu and could be either enumerated or temporal).

Clearly the degree of overlap between the items is a function of the particular set of items, and therefore the confusability of implicit spelling will vary from

[6]Where I grew up in Chicago we were in the **ED**gewater exchange, although this convention is no longer followed.

Ex.1018 / Page 138 of 343
AMAZON.COM, INC.



Figure 6.3. The labels on a telephone keypad.

database to database. But even explicit spelling can encounter collisions between choices. Although the login IDs mentioned above may be unique on a particular computer system, for other kinds of databases uniqueness is less likely. There may be many "Smiths" in a large company, and there are quite a few "Cambridge" streets in the Greater Boston area, for example. Whenever this occurs, an alternative selection will have to be provided regardless of the data input method.

Implicit spelling may be very effective. For many databases, confusability due to letter-to-key mapping is less significant than collisions due to duplicated spellings. For example, Davis [Davis 1991] reported on two databases of names of job applicants and students at a university. In one list of approximately 9,000 individuals, 40% of them shared a last name with at least one other individual, but only 8% of all names collided with another name using implicit spelling. A second list of 25,000 individuals exhibited 38% shared names with 21% of the names colliding.

Choices	Touch tone spelling
Brown	27696
Davis	32847
Green	47336
Jones	56637
Smith	76484
South	76884

Figure 6.4. A menu of last names and the associated touch tone spelling.

Ex.1018 / Page 139 of 343
AMAZON.COM, INC.

For either numeric or alphabetic input, termination of the input sequence may also be implicit or explicit. If the input is of a fixed length, e.g., the three digits of an area code, no terminator is necessary. If the input is of variable length, input may be terminated by a specific key, the system may detect that the user is finished by a pause, or the system may automatically complete the input as soon as a unique choice is identifiable.

Implicit termination requires a timer. The system asks a user for input and gathers digits until the user has paused for a predetermined period of time. The time-out may be longer during the period before the user has entered any digits to provide for thinking time. In the event of a time-out without any input, a help mechanism should be triggered. **Explicit termination** requires the use of a key that cannot be confused with user input. The "#" key is often used for this purpose.

Explicit termination requires more explanation to the user than does implicit termination and also requires that one remembers what to do at the end of input. Explicit termination also requires a timer as it must cope with the user who inputs the requested data but then forgets the terminator. Explicit termination is faster if the user does remember the terminator as the time-out period is not necessary. Explicit termination can also accept multiple, sequential fields of input more easily than implicit termination, such as the house number and street name of an address as the user can type each in sequence indicating segmentation of the items with the terminator key. For example, I could enter "E 1 5 * 3 2 7" as my building and room number without waiting to hear a prompt to enter my room number. To be more general, explicit termination allows type ahead; if the user knows what the system will ask next he or she can start to enter it without waiting for the time-out.

Because the input is of variable length, both input techniques should perform range checking on the input if possible (for example, the number of digits in a valid house number) and warn the user when errors are detected. It may be that the user forgot the delimiter and started to enter the next field.

Hybrid termination schemes are possible. One example is **optional termination;** if the user enters the termination key, then the input is complete; otherwise input is ended by a time-out. This approach allows the experienced user to save time, while not confusing the novice about the use of a terminator. Such a scheme is used for dialing the number for an international telephone call, since there is considerable variation in the length of international telephone numbers. The local telephone switch does not know how many digits to collect before connecting to the destination country and therefore must wait for the caller stop dialing, using a time-out. The caller can terminate the dialed digits with a "#" to indicate the end of input, avoiding the delay caused by the time-out.

Another hybrid approach allows automatic completion of user input when a unique choice is made; this saves keystrokes for the user. In the simplistic database illustrated in Figure 6.4, for example, the first digit will uniquely select one of the first four names but "Smith" and "South" require three digits for a unique choice. With automatic completion the system responds by echoing the selected choice as soon as it can be determined. But the user may be in the pro-

Ex.1018 / Page 140 of 343
AMAZON.COM, INC.

cess of entering the next digit(s) anyway so the system must quickly determine whether the additional input is a continuation of the spelling or the next step in the user interaction. One technique reported in [Gould *et al.* 1987] is to "absorb" this further input as long as it conforms to the remaining digits in the selected item.

Another shortcut for selecting alphabetic choices by spelling is user-initiated completion. A special key is used to jump from spelling input to a menu style selection. For example, to continue with the name database example, the user could type "3*" and get "Davis," which is unique. But entering "7*" or even "76*" would invoke a menu to select between "Smith" and "South." With user-initiated completion, the user gets to choose between pressing more digits in trade for a shorter menu, but with a large database it may not be intuitive how many names will match a sequence of a few digits.

CASE STUDIES

This section describes three voice response applications that illustrate application of the design considerations and interaction techniques just mentioned. Three different systems all employing speech synthesis for output are discussed. Both the first and third examples are telephone-based information retrieval systems, one for receiving driving directions and the other for reading electronic mail. The remaining system also gives driving directions but operates in real time in an automobile.

Direction Assistance

Direction Assistance [Davis and Trobaugh 1987] is a program that gives driving directions over the telephone.[7] It is an example of the usefulness of speech synthesis in situations where recorded speech would be much more difficult to employ for two reasons. First, Direction Assistance generates fairly complicated textual descriptions of an abstracted data structure, i.e., a route traversing links in a street database. Second, the database of street names is large enough that preparing a recording of each would be tedious. The database does accommodate phonetic spellings for those street names that the synthesizer mispronounces.

To use Direction Assistance, a user calls and enters both source and destination addresses with touch tones. The system then determines a route between these two points and translates the route into a textual description to be read by the speech synthesizer. The motivation for a telephone-based interface is to make this service available while driving around a city and, in particular, when one is

[7]Direction Assistance was written by Jim Davis and Thomas Trobaugh initially at Thinking Machines, Inc., and later at the MIT Media Lab. Davis also used it as a basis for work in intonation at AT&T Bell Laboratories [Davis and Hirschberg 1988].

Ex.1018 / Page 141 of 343
AMAZON.COM, INC.

lost. At such times, access to a computer terminal is unlikely.[8] Spoken directions can be sufficiently effective; a study by Streeter [Streeter *et al.* 1985] found recorded spoken directions to be more easily followed than either written directions or annotated maps.

Direction Assistance includes a street database covering 45 square miles of the Greater Boston area roughly centered on M.I.T. (see Figure 6.5). This map includes portions of a number of the municipalities surrounding Boston. (In Boston several common street names occur in multiple towns on different streets, which is a source of trouble.) This street database has been augmented with information such as street quality, locations of traffic lights and stop signs, and selected major landmarks. The database also notes some other geographic entries, including the locations of traffic circles, railroad tracks, and bridges.

The software components to satisfy these design goals are quite distinct and operate serially. Figure 6.6 shows these program segments. The first part of a user interaction is with the Location Finder, which has the job of eliciting from the caller his or her present location and desired destination. The Route Finder then consults the database to find a path to the destination; this path consists of a series of street segments found in the database. The Describer generates a textual description of the route describing it in terms of the actions a driver must take to follow the path. This text is then read to the user with appropriate pauses by the Narrator.

User Input

The Location Finder handles all user input. It describes the use of the touch tone keys on the telephone and explains that the "*" key is always available for help. It then asks the user for his or her location. All of the prompts are interruptible and type ahead is enabled for all except the yes-or-no questions. The input buffer is cleared whenever the system detects an error, and the error message is *not* interruptible; this is so the error message will not be lost due to type ahead.

In some versions of Direction Assistance, the Location Finder had access to a database linking telephone numbers to street addresses; such databases are commercially available. This access would be particularly useful if the caller were lost and calling from a pay phone, since in most states pay phones are required to display the phone number and address. As street signs are often missing in Boston, it is easy to get into a situation where a driver has no idea which street he or she is on. In telephone number gathering mode, the Location Finder waited until seven digits were entered, looked them up, and pinpointed the location of the caller.

More relevant to our discussion of user input is address mode, in which a house number and street name must be entered. The system uses explicit termination

[8]A company in the San Francisco area installed direction giving machines in public places such as convenience stores, much like video games. These systems used visual menus for selecting a destination and printed detailed driving directions.

Ex.1018 / Page 142 of 343
AMAZON.COM, INC.