Using Articulatory Position Data to Improve Voice Transformation

Arthur Richard Toth

CMU-LTI-09-002

December 1, 2008

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Alan W Black, Chair
Richard Stern
Mosur Ravishankar
Simon King, University of Edinburgh

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was supported in part by the US National Science Foundation under grant number 00414675 "TRANSFORM: flexible voice synthesis through articulatory voice transformation."

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Keywords: Articulatory Position, Voice Transformation

For my parents, Georgia S. and Richard Toth.

Abstract

Voice transformation (also known as voice conversion or voice morphing) is a name given to techniques which take speech from one speaker as input and attempt to produce speech that sounds like it came from another speaker. One compelling argument for good voice transformation is that it reduces the difficulty in creating additional synthetic voices with new identities and styles once an existing voice has been created based on a full-sized corpus. There are further voice transformation applications for security, privacy, and assistive technologies.

Although current voice transformation techniques perform well in the sense that humans typically judge transformed speech to sound more like the target speaker than the source speaker, there is still room for improvement.

We investigate the use of articulatory position data to improve voice transformation. When a person speaks, motions of the articulators affect the shape of the vocal tract, which affects the produced sound. Recently, data that includes measurements of the positions of various articulators along with recordings of the produced speech has been made publicly available. This articulatory position data gives us new information about the production of speech and has already been used successfully to predict quantities such as Mel-frequency cepstral coefficients [Toda et al., 2004a]. Such data gives us a different source of information from typical features derived from speech signals and enables promising new approaches to voice transformation.

One of the current challenges of using articulatory position data is that it is difficult to collect, so little is available. In order for it to be useful for more than a few speakers, some strategy must be devised to estimate it for other speakers. We present a number of techniques to do this and demonstrate that they are plausible by showing that artificial estimates of articulatory positions can be used to improve phonetic feature predictions similar to actual articulatory positions. Then we proceed to the question of using articulatory position features for voice transformation. Modifying the voice transformation process and representation of the articulatory data enables us to show improvement according to an objective metric. Then we demonstrate that artificial articulatory position estimates can also be used to improve voice transformation for speakers for whom no articulatory position data has been collected, according to this same objective metric.

As we are attempting to improve voice transformation, we give further consideration to what this actually means. Although a number of objective and subjective tests have been used to judge voice transformation quality, the best way to evaluate it is still an open question. We present new subjective and objective measures for voice transformation and report the results and our observations.

Acknowledgments

Ironically, at the stage of education where you are supposed to be doing your most independent work, it becomes even more necessary to collaborate with other people to succeed. A lot of people have been helpful as I have worked on this document, and I would like to give them my thanks.

I would like to thank Alan Black, my advisor, for all the help he has given me while I have been learning to perform research. I would also like to thank my other committee members, Rich Stern, Mosur Ravishankar, and Simon King, for their support as I have worked on my thesis.

I am also grateful for the collaboration Alan and I have had with Qin Jin and Tanja Schultz over the last year-and-a-half. Some of the results of our work together appear in this document.

As a student at CMU, I have interacted with many students and faculty members and am thankful for the friendships I have made and the knowledge we have shared. I am afraid it would be too hard to list everybody without leaving somebody out. In particular, I would like to thank Craig Olinsky for all the discussions we had while we were in the masters program and for challenging me to find research topics that better matched my interests. I would also like to thank Paul Placeway for advice and encouragement.

I have been fortunate to have good officemates during my time at CMU and have benefited from many conversations with them. I would like to thank Stefanie Tomko, Ananlada Chotimongkol, Thomas Harris, Tina Bennett, Antoine Raux, Kishore Prahallad, and Gopala Anumanchipalli for the times we shared. I would also like to thank Xiaojin (Jerry) Zhu, Brian Langner, and John Kominek. Though they were not my officemates, we were group members together, and I enjoyed working with them.

Many of the people I have already mentioned belonged to the Sphinx Group, which I would also like to thank for introducing me to many ideas and points of view.

I am also grateful for the support of my family and friends during this long Ph.D. process. I would like to thank my sister, Gabrielle Toth, and my nephew, Laszlo Toth, along with Frank Angotti, Jim Waugaman, John Li, Nick Batko, and others who participated in my experiments.

Finally, I would like to thank anyone who reads this document for showing interest in my work.

Contents

1 Introduction 1

1.1 Basic Terminology 1

1.2 Problem 2

1.3 Scope and limitations of investigation 2

1.4 Thesis Statement 3

1.5 Contributions 3

1.6 Thesis Overview 4

2 Speech Models 7

2.1 Introduction 7

2.2 Linear Prediction 9

2.2.1 Features 9

2.2.2 Analysis 10

2.2.3 Synthesis 14

2.2.4 Line Spectral Frequencies 15

2.3 Fourier Analysis-Based Speech Models 16

2.3.1 Fourier Analysis 16

2.3.2 Channel Vocoder 19

2.3.3 Phase Vocoder 21

2.3.4 Harmonic Sinusoidal Models 23

2.3.5 Multiband Excitation Vocoder 24

2.3.6 Harmonic Plus Noise Models 25
2.4 Cepstral Analysis 28
2.4.1 LPCC 29
2.4.2 MFCC 29
2.4.3 Mel Log Spectral Approximation Filter 30
2.5 Summary 30

3 Voice Transformation 31
3.1 Introduction 31
3.2 Applications 32
3.3 Related Voice Transformation Work 33
3.3.1 VQ Codebook-Based Techniques 34
3.3.2 GMM Mapping-Based Techniques 35
3.4 Festvox Voice Transformation Scripts 41
3.4.1 Training Process 41
3.4.2 Test Voice Transformation 46
3.5 Evaluation 47
3.5.1 Qualities 47
3.5.2 Measures 47
3.6 Summary 49

4 Using Articulatory Position Data with Voice Transformation 51
4.1 Introduction 51
4.2 MOCHA Database 52
4.3 Baseline Voice Transformation 54
4.3.1 Error Measure 54
4.4 Adding Articulatory Position Data 56
4.5 Experiments 57
4.5.1 Baseline Experiments 57

4.5.2 Attempts to Remove Noise from the Data 57
4.5.3 Attempts to Refine the Transformation Process 59
4.5.4 Representation of EMA Features 61
4.6 Summary and Conclusions 64

5 Cross-Speaker Articulatory Positions 67
5.1 Introduction 67
5.2 Acoustic, Articulatory, and Phonetic Features 68
5.3 Predicting Phonetic Features from Articulatory Positions 69
5.3.1 Phonetic Features 69
5.3.2 Articulatory Position Features 69
5.3.3 Model 69
5.3.4 msak0 Phonetic Feature Prediction Results 70
5.4 Cross-Speaker Articulatory Positions 72
5.4.1 Corpora 72
5.4.2 Cross-Speaker Mapping Approaches 72
5.4.3 Cross-Speaker MCEP/Articulatory Position Results 73
5.5 Cross-Speaker Phonetic Feature Prediction 75
5.5.1 fsew0 Phonetic Feature Prediction 75
5.5.2 FAF Phonetic Feature Prediction 77
5.6 Discussion 78
5.7 Voice Transformation with Cross-Speaker Articulatory Position Features . 78
5.7.1 Data 79
5.7.2 Features 79
5.7.3 Results 79
5.8 Conclusions 80

6 Evaluation of Voice Transformation 83
6.1 Introduction 83

6.2 Measuring Voice Transformation 83
6.3 Subjective Measures 84
6.4 Listening Experiment Design 84
6.5 Data Analysis 85
6.6 Transformation Triangle Diagrams 86
6.7 Evaluating Voice Transformation with TTDs 90
6.8 Subjective Measures Discussion 90
6.9 Automatic Speaker Identification for Measuring VT 92
6.10 Measuring Identity 93
6.11 Speaker Identification Systems 93
6.11.1 GMM-based SID System 93
6.11.2 Phonetic SID System 93
6.12 Data 94
6.13 Speaker Identity Experiments 95
6.13.1 Recorded Speech 95
6.13.2 Single-Model Experiments 95
6.13.3 Dual-Model Experiments 97
6.14 Experiments With Measuring Identity 97
6.14.1 CLUSTERGEN Synthesizer 99
6.14.2 Cluster Unit Selection Synthesizer 99
6.15 Objective Measure Discussion 100

7 Conclusions **103**
7.1 Summary 103
7.2 Contributions 104
7.3 Future Work 104

Bibliography **105**

List of Figures

3.1 Baseline Voice Transformation Training Process 42

3.2 Baseline Voice Transformation Training Process 43

3.3 Rule Used for Dynamic Time Warping 45

3.4 Goals of Various Speech Applications 48

4.1 *Transformation of Articulatory and Speech Data* 52

4.2 Articulatory Positions Recorded in the MOCHA Database 53

4.3 *Voice Transformation Training* 54

4.4 *Voice Transformation* 55

6.1 Similarities by Speaker Pair 86

6.2 Similarities by Knowledge of Speaker 87

6.3 Transformation Triangle Diagram Example 1 88

6.4 Transformation Triangle Diagram Example 2 88

6.5 Transformation Triangle Diagram Example 3 89

6.6 Transformation Triangle Diagram Example 4 89

6.7 Transformation Triangle Diagrams by Speaker Pair 91

6.8 Confusion Matrix for Single-Model Experiments with Transformed Speech and GMM-based SID 96

6.9 Confusion Matrix for Single-Model Experiments with Transformed Speech and Phonetic SID 96

6.10 Normalized GMM-based SID Scores for Various Types of Speech. Smaller values represent being closer to the target speaker and are better. “Real” is recorded speech, “VT” is voice transformed speech, “CG” is CLUSTERGEN synthetic speech, and “CL” is cluster unit selection synthetic speech. 98

6.11 Normalized Phonetic SID Scores for Various Types of Speech. Smaller values represent being closer to the target speaker and are better. “Real” is recorded speech, “VT” is voice transformed speech, “CG” is CLUSTERGEN synthetic speech, and “CL” is cluster unit selection synthetic speech. 98

List of Tables

4.1 *MCEP vs. EMAMCEP MCD Means (Std. Devs.)* 57

4.2 *VT (fsew0 to msak0) with EMA: Noise Compensation: MCD Means (Std. Devs.)* 59

4.3 *VT (msak0 to fsew0) with EMA: Noise Compensation: MCD Means (Std. Devs.)* 59

4.4 *VT (fsew0 to msak0) with EMA: Process Changes: MCD Means (Std. Devs.)* 60

4.5 *VT (msak0 to fsew0) with EMA: Process Changes: MCD Means (Std. Devs.)* 60

4.6 *VT (fsew0 to msak0) with EMA: Derived Features: MCD Means (Std. Devs.)* 62

4.7 *VT (msak0 to fsew0) with EMA: Derived Features: MCD Means (Std. Devs.)* 62

5.1 *msak0 Binary Phonetic Feature Prediction F-scores* 71

5.2 *Cross-Speaker MCEP/Articulatory Position Mappings* 73

5.3 *Cross-Speaker Mappings vs. Single-Speaker Mappings* 74

5.4 *fsew0 Binary Phonetic Feature Prediction F-scores* 75

5.5 *FAF Binary Phonetic Feature Prediction F-scores* 77

5.6 *Pseudo Lip Distance MCD Means (Std. Devs.)* 80

5.7 *Pseudo Lip Distance MCD Means (Std. Devs.)* 80

Chapter 1

Introduction

1.1 Basic Terminology

Voice Transformation (also called **Voice Conversion** or **Voice Morphing** by some) is a name given to techniques which take speech from one speaker as input and attempt to produce speech that sounds like it came from another speaker. One particularly compelling argument for good Voice Transformation is that it reduces the difficulty in creating synthetic voices with new identities and styles. Once a full-sized corpus has been collected from a source speaker, the amount of additional data needed to produce a new voice is much smaller than what is necessary to produce a new voice based on typical concatenative synthesis techniques alone.

Voice transformation is a branch of **speech synthesis**, which is concerned with the automatic production of speech. As speech is a complicated phenomenon, it has proven useful to analyze it from the perspective of **speech models**, which are parametric representations of features which are hypothesized to be important to the task at hand. The typical voice transformation process is thus based on the source and target speakers' parameters according to an underlying speech model and a mapping between these parameters. In the case of voice transformation, the goals for the underlying speech model are to produce speech that is natural, intelligible, and has the identity of the target speaker.

1.2 Problem

Although current voice transformation techniques appear to perform well in the sense that humans typically judge transformed speech to sound more like the target speaker than the source speaker, there is still room for improvement.

We propose the use of articulatory position data to improve voice transformation. When a person speaks, motions of the articulators affect the shape of the vocal tract, which affects the produced sound. Recently, data that includes measurements of the positions of various articulators along with recordings of the produced speech has been made publicly available. This articulatory position data gives us new information about the production of speech and has already been used successfully to predict quantities such as Mel-frequency cepstral coefficients and acoustic-phonetic features [Toda et al., 2004a],[Toth, 2005]. Such data gives us a different source of information from typical features derived from speech signals and enables new approaches to voice transformation.

As we are attempting to improve voice transformation, we will need to consider what this actually means, Although a number of objective and subjective tests have been used to judge voice transformation quality, the best way to evaluate it is still an open question. For this reason, we will also be investigating methods of voice transformation evaluation.

1.3 Scope and limitations of investigation

Currently, there are a number of techniques used to perform voice transformation. The ones investigated in this thesis are based on Gaussian Mixture Model mappings whose parameters are learned from “parallel” speech examples, where there are acoustic waveforms of both speakers reading the same text. This is currently the most prominent approach to voice transformation, though some techniques based on frequency warping [Erro et al., 2008] and Hidden Markov Model speaker adaptation [Zen et al., 2007] are gaining popularity.

At this point, voice transformation techniques focus primarily on transforming the acoustic characteristics between speakers, with some effort to modify short-term prosodic features. This appears to mainly be due to our better knowledge of how to model these lower-level speech quantities and difficulties with data sparsity that would occur with higher-level features that span greater time intervals. Our work continues along these lines and does not address the transformation of higher-level features such as prosodic contours (except incidentally through local modifications) or word choice.

A number of methods have been used to record articulatory position data. This thesis only uses articulatory data that was collected using an **Electro-Magnetic Articulograph (EMA)** on continuous read speech [Wrench, 1999]. There is other existing data that was created using X-ray technology [Lenzo and Fujimura, 2001], but it was often recorded for single (and sometimes nonsense) words, and is less appropriate for our work. Furthermore, only a small amount of it is available.

1.4 Thesis Statement

Over time, the dominant speech synthesis techniques have progressed from physical and acoustic models to data-driven methods, This has enabled the creation of synthetic voices that sound more natural and have more recognizable identities, but the flexibility of such systems is limited by the amount and type of collected data. Now that there has been some success with data-driven techniques, there is a desire to reincorporate some of the flexibility of model-based techniques in an effort to reduce the amount of necessary data. Articulatory positions present a tangible way to help parametrize speech data that differs from traditional methods that are based on analysis techniques from Digital Signal Processing (DSP) applied to acoustic waveforms. In addition to allowing the modification of parameters in a different space, articulatory positions are also subject to static and dynamic physical constraints that can be incorporated into models of speech to make them more realistic. New methods for analyzing speech based on modeling of articulatory positions can be used to improve voice transformation. Voice transformation can be improved in terms of intelligibility, naturalness of speech, and identity of speaker. Also, new evaluation techniques can be created that provide more insight into the quality of voice transformation.

1.5 Contributions

The primary contributions of this work are:

1. an investigation of using articulatory position data to improve voice transformation,

2. improvement of voice transformation, and

3. new subjective and objective evaluation techniques for voice transformation.

1.6 Thesis Overview

Chapter 2 describes speech models and provides descriptions of some of the most common models used for voice transformation. It is essentially a literature survey focused on the historical line of models that led to the current ones that are popular in voice transformation. The popular speech model techniques are divided into three main categories: linear prediction, Fourier analysis, and cepstral analysis. Using these basic techniques, a range of speech models is described.

Chapter 3 describes voice transformation based on speech models discussed in Chapter 2. It is essentially a literature survey that focuses on the historical line of voice transformation models that led to the ones used in this thesis. It includes an in-depth description of the baseline voice transformation system that is the point of departure for our further experiments. The historical development of "linear" voice transformation is traced from Abe's early work with codebooks through the GMM-based mappings of Stylianou, Kain, and Toda.

Chapter 4 describes articulatory position data and attempts to use it to improve the voice transformation system described in Chapter 3. A range of experiments from our SSW6 paper [Toth and Black, 2007] is described here. They begin with straightforward extensions of the feature vectors to include articulatory position data and then present various modifications to improve voice transformation according to the objective **Mel-Cepstral Distortion (MCD)** metric. The modifications included attempting to remove noise from the data, attempting to modify the transformation procedure when certain steps seemed less appropriate for the additional features, and using derived features based on a combination of articulatory positions. The final result was a very small positive result when a combination of these techniques was used.

Chapter 5 describes using articulatory position data from one speaker with another speaker. It is currently difficult to collect articulatory position data, so if the small amount that is available is to be useful, there must be strategies to make it more generally helpful. The first part of this chapter is based on our Interspeech 2005 paper [Toth, 2005]. First, it presents results on using articulatory position data to predict phonetic features, and then it demonstrates that using articulatory position data can improve performance over traditional spectral features. Next, cross-speaker articulatory position predictions, which are articulatory position estimates for speakers, are investigated and also show the ability to improve phonetic feature performance over traditional spectral features. After this demonstration of the viability of cross-speaker articulatory positions for phonetic feature prediction, the second part of the chapter concerns itself with their application to voice transformation.

Chapter 6 describes attempts to measure voice transformation using **Speaker IDentification (SID)** systems. There is a discussion of how measuring identity, along with naturalness and intelligibility, is important in the evaluation of voice transformation. There is a discussion of how speaker identification systems can be used to measure the identity of transformed speech and a description of two different speaker identification systems that we used for this task: a GMM-based systems and phonetic system. The rest of the chapter is based on experiments from our ICASSP 2008 paper [Jin et al., 2008]. There is a description of data used for the experiments followed by attempts to fool the SID systems with VT in a couple contexts. then there is a description of our novel approach to using the SID scores as a metric for VT and a comparison to the scores for recorded speech and two types of synthesizers, which are also described.

Chapter 7 is the conclusion which summarizes the results and discusses the contributions. It also discusses future directions, placing this work in a broader context.

Chapter 2

Speech Models

2.1 Introduction

A **speech waveform** is a function of sound pressure versus time that describes speech. Although speech waveforms are typically thought of as continuous, their computer representations consist of discrete samples. This is not a problem as a sufficiently high sample rate combined with a sufficiently large quantization space captures as much information as is necessary to enable production of a sound that is indistinguishable from the original speech waveform to the human ear. This is possible due to limitations in the range of frequencies that humans can hear and in their ability to distinguish amplitude levels. **Amplitude** is the amount of sound pressure in the speech waveform at a particular point in time.

When representing speech in computational algorithms, there are factors which can make the discretely sampled waveform an inconvenient representation. Speech is produced through a physical process that combines numerous factors that contribute in different ways to the final waveform. The speaker's lungs, vocal folds, velum, tongue, teeth, and lips all influence the speech waveform in their own ways. Speech is perceived through another physical process that is based on other factors. The listener's ears respond to different parts of the speech waveform based on their anatomy. The connection between these factors and speech waveforms is often complicated, and computation is necessary to determine or estimate them. In speech synthesis, it is desirable to represent these processes and their contributions to the production or perception of the speech waveform, because it provides the ability to produce and modify new speech waveforms based on these processes.

A **speech model** is a way of representing speech based on quantities called **features** and an interpretation of those features. The process of producing values for a speech model's features from a waveform is called **analysis**, and the process of producing a waveform from the values of a speech model's features is called **synthesis**. It should be noted that this use of the word synthesis should not be confused with **speech synthesis**, which does produce waveforms, but can include numerous additional processes, such as textual analysis. Synthesis, in the speech model sense, is often the last step of a speech synthesis process.

Earlier attempts to find suitable representations for speech were often based on considerations of reducing the size of the speech data for more efficient transmission. These techniques are part of the field of **speech coding**. Although reducing the size of speech data is often desirable in speech synthesis, it usually has lower priority than properties such as the naturalness or intelligibility of the synthesized speech. Having a compact representation of speech, however, still has benefits in that it can be more efficient, and that it may lead to fewer problems with data sparsity if the speech model data is used to train a statistical model. The difference in goals and emphases in speech synthesis led to further refinements of speech coding techniques and even new techniques.

As this work is concerned with the sub-discipline of speech synthesis called **voice transformation**, the rest of this chapter will describe some of the main speech models that have been used in its implementation. Voice transformation, itself, will be the topic of Chapter 3. For this chapter, it is sufficient to note that, in addition to the typical speech synthesis goals of naturalness and intelligibility, voice transformation has the goal of accurately representing a speaker's identity. These goals have influenced the choices of speech models used for voice transformation. At this point, the most prevalent speech models used for this task are based on **linear prediction**, **Fourier analysis**, and **cepstral analysis**, which will be described in Section 2.2, Section 2.3, and Section 2.4 respectively. These sections will contain some general information about the techniques and descriptions of representative speech models that are commonly used in voice transformation.[1] In practice, some of the speech models use more than one of these techniques, so there is some level of arbitrariness in where they can be described. In most cases, however, one technique is used more heavily than the others. The descriptions of the speech models will include their sets of features and the analysis and synthesis processes used with them. Some details from signal processing will be included in these descriptions, but a full explanation of the background signal processing theory is beyond the scope of this

[1] As these sections describe work by other people in detail, for brevity, the references have been mentioned once near the beginnings of the sections. The techniques and equations are assumed to be from these primary references unless explicitly specified otherwise.

document.

2.2 Linear Prediction

According to Markel and Gray Jr. [1976], linear prediction was first applied to speech by Saito and Itakura [1966] and by Atal and Schroeder [1967], while some of the underlying mathematics were performed by Gauss as early as 1795 [Sorenson, 1970]. The underlying assumption of **linear prediction** is that each element of a sequence of values can be predicted as a **linear combination** of a finite number of preceding values. The number of preceding values used in the linear combination is known as the **order** of the linear predictive model. In equation form, a linear predictive model looks as follows:

$$\tilde{s}[n] = \sum_{p=1}^{P} a_p s[n-p] \tag{2.1}$$

where $s[n]$ is the value of the nth sample of the signal, in this case a speech waveform; $\tilde{s}[n]$ is an estimate of $s[n]$; P is the order of the analysis; and the a_ps are coefficients that are selected during analysis.

2.2.1 Features

When processing speech, it is common to treat it as a number of **frames**, which are portions of speech that are short enough to capture the local phenomena being studied. For each frame of speech that is represented by a linear prediction model, the features include a set of a_p coefficients as in Equation 2.1, and an **excitation**. The excitation is an input signal that is added to the linear combination in Equation 2.1. It can be considered an input signal to the model. A non-zero excitation is necessary for a linear predictive model to produce non-zero output. The excitation, itself, may be represented in a number of ways. Three popular choices are:

1. a gain estimate and a voiced/unvoiced decision with pitch period estimates for voiced regions of speech, which can be expanded by rules to create a signal

2. a voiced/unvoiced decision with multiple pulses for each voiced region

3. a residual or error signal that is derived after solving for the a_p coefficients.

2.2.2 Analysis

One common goal among different variations of linear predictive analysis is to find values for the a_p coefficients that produce the most accurate estimates of the original speech signal. One way to formulate this problem is in terms of minimizing the squares of the **error**, which is the difference between the original signal, $s[n]$, and the estimates, $\tilde{s}[n]$,over a range of samples. Two popular methods of selecting coefficients, based on minimizing squared error, are **the autocorrelation method** and the **the covariance method**. A third popular method, which belongs to a class of **lattice methods** based on a different set-up, is used to produce **partial correlation**, **PARCOR**, or **reflection** coefficients. This particular lattice method is solvable in a manner similar to the autocorrelation method.

Minimizing Squared Error

The squared error can be minimized using the typical procedure from calculus of equating derivatives with zero and solving. In the specific case of a linear predictive model, where we are considering a range of samples indexed by the variable, m, we get an equation for each coefficient, a_i, where $1 \leq i \leq p$:

$$\frac{\partial}{\partial a_i}\sum_m (s[m] - \tilde{s}[m])^2 = 0 \tag{2.2}$$

$$\frac{\partial}{\partial a_i}\sum_m \left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right)^2 = 0 \tag{2.3}$$

$$\sum_m \frac{\partial}{\partial a_i}\left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right)^2 = 0 \tag{2.4}$$

$$\sum_m 2\left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right)\frac{\partial}{\partial a_i}\left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right) = 0 \tag{2.5}$$

$$\sum_m 2\left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right)(-s[m-i]) = 0 \tag{2.6}$$

$$\sum_m \left(s[m] - \sum_{p=1}^{P} a_p s[m-p]\right) s[m-i] = 0 \tag{2.7}$$

$$\sum_m s[m-i]s[m] = \sum_{p=1}^{P} a_p \sum_m s[m-i]s[m-p] \tag{2.8}$$

At this point, it is typical to define a function, ϕ, as follows:

$$\phi(i,p) = \sum_{m} s[m-i]s[m-p] \tag{2.9}$$

and to rewrite Equation 2.8 as:

$$\sum_{p=1}^{p} a_p \phi(i,p) = \phi(i,0) \tag{2.10}$$

This resulting system of equations from considering $1 \leq i \leq p$ can be represented in matrix form as follows:

$$\begin{pmatrix} \phi(1,1) & \phi(1,2) & \phi(1,3) & \cdots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \phi(2,3) & \cdots & \phi(2,p) \\ \phi(3,1) & \phi(3,2) & \phi(3,3) & \cdots & \phi(3,p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi(p,1) & \phi(p,2) & \phi(p,3) & \cdots & \phi(p,p) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \phi(1,0) \\ \phi(2,0) \\ \phi(3,0) \\ \vdots \\ \phi(p,0) \end{pmatrix} \tag{2.11}$$

This equation can be written more compactly by substituting variable names for the corresponding matrix and vectors as follows:

$$\Phi a = \psi \tag{2.12}$$

Up to this point, the range of the index of summation, m, has not been specified. One possibility may be to let m range over an entire speech waveform, but that would provide information on the entire waveform and would not give any insight into the local variations in the waveform. When performing linear prediction analysis, it is typical to consider smaller segments of a waveform in separate analyses. This leads to different choices for m, which lead to different solutions of the system of equations represented by Equation 2.12.

Autocorrelation Method

In the **autocorrelation method**, the index of summation, m, is chosen to range from $-\infty$ to ∞, but the waveform, $s[n]$, is assumed to be zero when $n < 0$ or $n \geq N$ for some fixed value N. Given this assumption, it can be shown that $\phi(i,j) = \phi(k,l)$ when

$|i-j| = |k-l|$. As a result, the Φ matrix in Equation 2.12 can be written as:

$$\Phi = \begin{pmatrix} \phi(1,1) & \phi(1,2) & \phi(1,3) & \cdots & \phi(1,p) \\ \phi(1,2) & \phi(1,1) & \phi(1,2) & \cdots & \phi(1,p-1) \\ \phi(1,3) & \phi(1,2) & \phi(1,1) & \cdots & \phi(1,p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi(1,p) & \phi(1,p-1) & \phi(1,p-2) & \cdots & \phi(1,1) \end{pmatrix} \tag{2.13}$$

The resulting Φ matrix can be seen to be **Toeplitz**, which means that along any (upper-left to lower-right) diagonal, all the values are the same. Furthermore, this matrix is **symmetric**. The value in any position is equal to the value in the position with the row and column swapped. Due to these constraints on the Φ matrix, Equation 2.12 can be solved using an efficient algorithm called **Levinson-Durbin recursion**. This procedure only requires on the order of p^2 operations to solve for the a vector, where a typical approach involving a general matrix inversion would require on the order of p^3 operations.

Covariance Method

In the **covariance method**, the index of summation, m, is limited to the range of $0 <= m < N$ for some fixed value N, but the speech waveform values outside this range are used when necessary, and not set to zero as in the autocorrelation method. Given this assumption, it can be shown that $\phi(i,k) = \phi(k,i)$. As a result, the Φ matrix in Equation 2.12 can be written as:

$$\Phi = \begin{pmatrix} \phi(1,1) & \phi(1,2) & \phi(1,3) & \cdots & \phi(1,p) \\ \phi(1,2) & \phi(2,2) & \phi(2,3) & \cdots & \phi(2,p) \\ \phi(1,3) & \phi(2,3) & \phi(3,3) & \cdots & \phi(3,p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi(1,p) & \phi(2,p) & \phi(3,p) & \cdots & \phi(p,p) \end{pmatrix} \tag{2.14}$$

The resulting Φ matrix is symmetric and **positive definite**, which means that for any nonzero vector (of appropriate length), v, $v^T \Phi v > 0$. Due to the special form of the Φ matrix in this case, an algorithm called **Cholesky decomposition** can be used to simplify the procedure of solving for the a vector. This usually takes about half the time of an approach involving a general matrix inversion, but still requires on the order of p^3 operations, so it is usually less efficient than the Levinson-Durbin recursion, which is used in the autocorrelation method.

Lattice Methods

Lattice methods include a number of techniques that formulate the problem of minimizing the error by solving for values in a structure known as a **lattice filter**. Although this general approach includes solutions which optimize quantities other than the squared error, the commonly used approach that produces **reflection coefficients** through **partial correlation** or **PARCOR** analysis is actually equivalent to the autocorrelation method, and its parameters, though different from autocorrelation parameters, can also be produced through Levinson-Durbin recursion. Thus, this commonly used form of lattice method can also be performed on the order of p^2 operations.

Excitation Parameters

Regardless of the method used to determine the a_p coefficients in Equation 2.12, it is still necessary to represent an excitation to represent the entire speech waveform.

The excitation used in a linear prediction model is $x[n]$ in the following equation:

$$\tilde{s}[n] = \sum_{p=1}^{P} a_p s[n-p] + x[n] \tag{2.15}$$

The calculation of the excitation parameters for each analyzed segment of speech depends on the choice of the excitation representation.

One popular model for the excitation is to treat it as a gain parameter, G, multiplied by a signal, $u[n]$, which is either a periodic train of unit impulses or white noise:

$$x[n] = Gu[n] \tag{2.16}$$

In this model, an external algorithm is necessary to provide decisions for whether the speech segment is voiced or unvoiced. An external algorithm is also necessary to estimate the **pitch period**, or number of samples after which the waveform approximately repeats, for segments that are judged to be voiced. The area of algorithms for judging voicing and estimating pitch periods is quite large and is a topic in itself [Hess, 1983]. Whatever methods are chosen, features will be necessary for voicing and pitch periods. The remaining feature in this model is gain, which can be estimated by the formula

$$G = \sqrt{R(0) - \sum_{p=1}^{P} a_p R(p)} \tag{2.17}$$

where the a_ps are the coefficients from Equation 2.12, and $R(k) = \sum_{m=0}^{N-1-k} s[m]s[m+k]$ [Rabiner and Schafer, 1978].

Another approach is to represent the excitation with a set of multiple pulses for each pitch period. The locations and amplitudes of the pulses are typically determined through an **analysis-by-synthesis** method [Atal and Remde, 1982].

A third approach that is popularly used in speech synthesis, due to the quality of the output is to set the excitation equal to the residual, or error signal, after determining the a_p coefficients in Equation 2.12:

$$x[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{p=1}^{P} a_p s[n-p] \tag{2.18}$$

Storing the residual requires one value for every sample in the original speech signal, but there are still cases where it is a useful representation of the excitation. It does allow recovery of the original signal, and if the linear prediction model matches the original signal well, the residual values will cover a smaller range and can most likely be compressed better than the original signal Press et al. [1992].

2.2.3 Synthesis

After linear predictive analysis has been performed on the segments of a speech waveform and coefficients have been extracted, synthesis consists of the following steps:

1. producing an excitation for each speech segment
2. filtering the excitation based on the linear prediction coefficients
3. combining the synthesized speech segments into one waveform

Producing an Excitation

The production of the excitation depends on the choice of representation. If, during analysis, it was treated as a gain multiplied by an impulse train or white noise, the following steps are taken:

1. Check the voicing decision

2. For voiced speech, generate an impulse train where the impulses are spaced according to the pitch period
3. For unvoiced speech, generate white noise.
4. Multiply the resulting signal (impulse train or white noise) by the gain.

If during analysis, the excitation was represented by multiple pulses or a residual, just produce those.

Filtering the Excitation

After the excitation, x[n], is produced, take the a_p coefficients and calculate the following equation over the samples, n, in the current speech segment:

$$\tilde{s}[n] = \sum_{p=1}^{P} a_p s[n-p] + x[n] \tag{2.19}$$

Combining Speech Segments

After speech segments are synthesized by, there are a number of choices for connecting them. One possibility is to simply concatenate them. This may lead to discontinuities where the segments are joined. Another possibility is to use some form of interpolation. This can be performed by synthesizing regions that are larger than the original analysis segments and using an overlap-and-add approach with appropriate windows.

2.2.4 Line Spectral Frequencies

During applications, such as voice transformation, it is necessary to predict speech model features, in some cases producing new values which weren't seen in the original data. Although linear prediction coefficients can effectively be used to model speech, small changes in the values of coefficients, even when they were derived from recorded speech, can lead to large changes in the quality of synthesized speech, even leading to a loss of **stability**, in the sense that the output of a linear prediction filter may no longer be a bounded function of the input. One way around this difficulty is to move the values of the linear prediction coefficients into a space that is easier to work with by converting them into **line spectral frequencies**, which are the complex roots of the following pair of equations:

$$P(z) = A(z) - z^{-K+1}A(z^{-1}) \tag{2.20}$$

$$Q(z) = A(z) + z^{-K+1}A(z^{-1}) \tag{2.21}$$

where K is the order of the linear prediction analysis, and $A(Z) = 1 + \sum_{k=1}^{K} a_k z^{-k}$. The line spectral frequencies have numerous convenient properties. They all lie on the unit circle. As the angle is increased while traveling around the unit circle, the roots of $P(z)$ and $Q(z)$ alternate. Stability is preserved when the angles are changed.

2.3 Fourier Analysis-Based Speech Models

The **harmonic sinusoidal** [McAulay and Quatieri, 1986] and **harmonic plus noise** [Stylianou, 1996] models are two speech models that have been used in voice transformation systems. They are based on a mathematical technique called Fourier analysis, and can be seen as two of the more recent developments in a line of speech models that goes back at least as far as Homer Dudley's channel vocoder [Dudley, 1939].

As a proper understanding of these speech models depends on knowledge of short-time Fourier analysis, some relevant background material on this topic will be provided first. It should be noted, however, that the fields of Fourier analysis and signal processing are large, and covering them in great detail is beyond the scope of this document.

2.3.1 Fourier Analysis

Fourier analysis is a powerful mathematical technique that has been used on a variety of problems including the modeling of speech signals. Among its numerous appealing characteristics are its ability to represent periodicities at different frequencies in signals. As the human ear uses a substructure called the cochlea to detect different frequencies, analyzing the frequencies in a speech waveform seems to be a natural way to extract information that people use to perceive speech.

Another compelling property of Fourier analysis is that an efficient algorithm for handling discrete-time signals in a discrete frequency sense, called the **Fast Fourier Transform (FFT)**, has been discovered. This, in turn, has allowed the creation and exploration of a number of more complicated speech models that are in part based on Fourier Analysis.

Fourier Transforms

Fourier analysis is based on different types of **Fourier Transforms**, which are specific types of mathematical mappings from one function to another. When handling discrete-time signals, where values are known only at certain fixed-time intervals, the **Discrete-Time Fourier Transform**, or **DTFT**, is used. If a discrete-time signal is represented by a function, $x[n]$, which maps an integer, n, to a complex value, the DTFT of $x[n]$ is:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{2.22}$$

where j is the imaginary unit, and ω is a frequency.

When the DTFT exists, the original signal can be retrieved from it through the relationship:

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \tag{2.23}$$

Although a signal can be represented in this way, it is not a convenient representation for speech synthesis. Taking the DTFT of a speech signal will provide information about the frequencies in the entire signal, which can be composed of many different types of sounds that have different frequency characteristics. Looking at all the frequencies present in the entire signal does not describe the relative powers at different frequencies at different times and how they change over time. This information is important for analyzing speech signals.

One type of Fourier Transform that can be used to analyze frequency information on shorter pieces of a signal, albeit with some drawbacks, is the discrete-time **Short-Time Fourier Transform (STFT)**. The discrete-time STFT is performed by applying a **windowing function**, which emphasizes local samples, to the DTFT. The discrete-time STFT for a signal $x[n]$ at sample, n, is:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} w[n-m]x[m]e^{-j\omega m} \tag{2.24}$$

where $w[n]$ is the window function. There are a variety of choices for the window, though it is typical to use one which has non-zero values only for a short interval around zero. Picking a specific window function can have a great effect on the properties of the discrete-time STFT, including whether the original signal can be recovered from it. If $w[0] \neq 0$, then the original signal can be recovered from the discrete-time STFT through the formula:

$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n,\omega) e^{j\omega n} d\omega \tag{2.25}$$

A fuller description of the considerations for selecting an appropriate window for the discrete-time STFT can be found in Chapter 6 of Rabiner and Schafer [1978]. The important thing to note is that it is possible to choose a window which will allow the original signal to be recovered exactly. In some sense, the representation can still contain all the information from the original signal.

Considering the points mentioned above, the discrete-time STFT may look like a good candidate for a speech model, and indeed numerous speech models have been based on it. Examining a few historical and prominent ones will give a sense of some of the issues involved. The first speech model that is arguably related to these principles is the **Channel Vocoder**. Though technically, it was based on bandpass filters and did not perform Fourier analysis, it influenced later speech models which did use Fourier analysis. Over 20 years later, it was followed by the **Phase Vocoder**, which attempted to correct some of its deficiencies.

Although the discrete-time STFT treats time discretely, it still treats signal values continuously. This was reasonable for older speech models based on analog hardware, but as digital computers became more powerful, it was typically more convenient to treat the signal values discretely as well. When both time and values are treated as discrete quantities, another transform called the **Discrete Fourier Transform (DFT)** is used:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{2.26}$$

where N is the number of points where the frequency is sampled. An associated discrete STFT is useful for analyzing signals with properties that vary over time and is built by applying a windowing function to the DFT:

$$X(n,k) = \sum_{m=-\infty}^{\infty} w[n-m] \sum_{n=0}^{N-1} x[m] e^{-j2\pi km/N} R_N(k) \tag{2.27}$$

Here, the function $R_N(k)$ equals 1 when $0 \leq k < N$ and 0 otherwise. [2]

[2] This R_N function is also implicitly in the DFT equation, which is only summed from 0 to $N-1$. For a fuller description of related issues for DFTs see Chapter 8 of Oppenheim et al. [1999]. For a fuller description of Short-Time Fourier Transforms, see Chapter 6 of Lim and Oppenheim [1988].

As time progressed, not only did computers become much more powerful, but a more efficient way of performing the DFT and its inverse, called the **Fast Fourier Transform (FFT)**, was discovered. This led to the construction of numerous software-based speech models. Some of the more prominent ones were **Multi-Band Excitation**, the **Harmonic Sinusoidal** model, and **Harmonic plus Noise Models (HNM)**. These models will be discussed in the following sections.

2.3.2 Channel Vocoder

In 1939, before computers were available, and long before the FFT was used on computers, Homer Dudley created the channel vocoder, which was the earliest speech coding device [Rabiner and Schafer, 1978],

Features

The channel vocoder's features are: voicing decisions, pitch estimates, and spectrum, which is in the form of magnitude outputs of bandpass filters in a finite set of frequency bands.

Analysis

The original channel vocoder was built from electrical circuits [Dudley, 1939]. Deriving features from speech waveforms meant producing new electrical signals from a signal representing the original speech waveform.

The circuitry to produce a pitch signal consisted of multiple steps. First, a frequency discriminator circuit greatly attenuated contributions at low frequencies, which were not plausible as fundamental frequencies for human speech, and progressively attenuated frequencies above a certain level so contributions in a plausible range for fundamental frequencies would tend to dominate the signal. Then, the output of this circuit was passed to a frequency meter circuit based on gas tubes that was developed by Hull [1929] [Hull, 1933] and modified by Riesz [Dudley, 1939]. The frequency meter circuit created a series of electrical pulses at a rate proportional to the detected frequency. Next, the frequency meter output is put through a low-pass filter to produce the final pitch signal.

Voicing is coded in the pitch signal. Regions without pulses are considered unvoiced.

Spectrum estimation is performed after a pre-distorting equalizer is applied to boost the contributions of higher frequency components. This is very much like the pre-emphasis filters used in many current speech analysis techniques. Then, the signal is split into 10 signal bands by passing it through 10 bandpass filters in parallel. The first bandpass filter is for frequencies from 0-250Hz, and the others each have bandwidths of 300Hz and are successively placed to cover up to a maximum frequency of 2950Hz.[3] The output of each bandpass filter is sent through a rectifier to measure the power (phase is not measured), and the resulting signal is low-pass filtered. The resulting signals are the spectral features.

A few interesting things to note about channel vocoder analysis are:

1. The idea of separating the pitch from the spectrum in speech, akin to the concept of source-filter models, was present at least as early as 1939.

2. The idea of using a filter to emphasize higher frequencies before spectral analysis, akin to pre-emphasis, was also preseent at least as early as 1939.

3. Combining pitch and voicing decisions into one stream, where a 0 pitch meant unvoiced speech was done at least as early as 1939.

4. The production of an explicit signal for pitch and voicing enabled easy pitch and voicing modification.

5. The spectral features did not attempt to measure phase.

Synthesis

The channel vocoder synthesizes speech by the following process. Part of the signal representing the pitch is sent to a relay which selects either a "buzz" source for voiced speech when there are pulses or a "hiss" source for unvoiced speech when there are no pulses. The remaining part of the pitch feature signal is sent through a relaxation oscillator which controls the frequency of the "buzz" source for voiced speech. One consequence is that the "hiss" source is used during silence, and the spectral features are needed to drive the output to zero in this case.

For each frequency band, the source signal is sent through a band-pass filter with the same pass-band as the one used during analysis.[4] For each frequency band, a balanced modulator is used to essentially multiply the band-limited source signal by the magnitude signal that is produced by the spectral analysis. The resulting parallel signals for the 10

[3] In later versions, different numbers and placements of bands were tried [Gold and Rader, 1967].

[4] Some later variations use smaller bandwidths in the synthesis band-pass filters than the ones used in the analysis band-pass filters in order to compress the signal[Gold and Rader, 1967]

bands are then combined into one signal, and a restoring filter is applied to undo the spectral distortion caused by the pre-distorting equalizer during the analysis of the spectrum.

In the opinion of Dudley [1939], intelligibility is most related to the spectrum and emotion is most related to the pitch.

According to Rabiner and Schafer [1978], some of the issues with the channel vocoder are that the synthesized speech is reverberant due to the way adjacent frequency bands are merged, formants can be highly distorted through quantization, and the system depends on pitch and voicing estimates, which are difficult to accurately predict.

2.3.3 Phase Vocoder

Flanagan and Golden [1966] introduced the Phase Vocoder in 1966. One of the main motivating factors was the belief that the synthesis quality from previous speech models could be improved through better handling of the excitation. Instead of explicitly tracking pitch and voicing like the channel vocoder, the Phase Vocoder extracts phase information from the signal, in terms of its derivative. At this stage in time, a computer was used to implement the Phase Vocoder, but the implementation was considered to be a simulation.

Features

The phase vocoder features are spectral amplitudes and phase derivatives.

Analysis

The amplitudes and phase derivatives are calculated at various frequency and time locations by first estimating quantities a and b, which amount to essentially the real and imaginary parts of the discrete STFT:

$$a(\omega_n, mT) = T\sum_{l=0}^{m} f(lT)[cos\omega_n lT]h(mT - lT) \tag{2.28}$$

$$b(\omega_n, mT) = T\sum_{l=0}^{m} f(lT)[sin\omega_n lT]h(mT - lT) \tag{2.29}$$

where ω_n is the nth frequency that is sampled, m is the number of the speech sample, T is the sampling interval of the speech signal, and h is a window function. From these quantities, the magnitudes, $|F|$ at the ω_n sample frequencies and mT sample times are calculated as:

$$|F(\omega_n, mT)| = (a^2 + b^2)^{\frac{1}{2}} \tag{2.30}$$

For convenience the arguments of a and b have been left out, but they match the arguments of $|F|$.

The discrete phase derivative estimates, $\frac{\Delta\phi}{T}$, at the same points are calculated as:

$$\frac{\Delta\phi}{T}(\omega_n, mT) = \frac{1}{T}\frac{b\Delta a - a\Delta b}{a^2 + b^2} \tag{2.31}$$

where the Δs are the change in function values with respect to a change in sample time, T.

$$\Delta a = a(\omega_n, (m+1)T) - a(\omega_n, mT) \tag{2.32}$$

$$\Delta b = b(\omega_n, (m+1)T) - b(\omega_n, mT) \tag{2.33}$$

Some of the original parameter choices were $T = 10^{-4}$ seconds, h was a sixth-order Bessel filter, the number of frequency channels was 30, and $\omega_n = 2\pi n(100)$ rad/sec. The ω_n values were chosen so the spectrum was analyzed from 50 Hz to 3050 Hz.

Synthesis

Synthesis is performed by synthesizing each frequency channel and adding all the synthesized channels together. Each individual channel, $\tilde{f_n}$, was synthesizing by performing the following calculation:

$$\tilde{f_n} = |F_n(\omega_n, mT)|cos\left(\omega_n mT + T\sum_{l=0}^{m}\frac{\Delta\phi(\omega_n, lT)}{T}\right) \tag{2.34}$$

According to Rabiner and Schafer [1978], speech synthesized by a phase vocoder can be reverberant due to a lack of absolute phase information. The reason its creators decided to use the phase derivative instead of the absolute phase was because the absolute phase was unbounded.

Numerous variations on the phase vocoder appeared in the literature, including a method of basing the calculations on the FFT [Portnoff, 1981].

One additional point about phase vocoder synthesis is that the model can be manipulated to modify time and frequency properties of the speech signal. These operations can be useful in speech synthesis.

2.3.4 Harmonic Sinusoidal Models

Sinusoidal coding, or harmonic sinusoidal modeling, was introduced by McAulay and Quatieri [1986]. It differs from models such as the channel vocoder and phase vocoder in that instead of using a set of fixed frequency bands to represent the speech signal, it uses a set of sinusoids whose frequencies are multiples of the fundamental frequency, which varies over time.

Features

The underlying assumption is that each frame, $s(n)$, of a speech signal can be represented adequately by the following expression:

$$s(n) = \sum_{l=1}^{L} A_l cos(nl\omega_0 + \phi_l) \tag{2.35}$$

where L is the number of harmonics, A_l is the amplitude of the lth harmonic, ω_0 is the fundamental frequency, and ϕ_l is the phase of the lth harmonic. These quantities at each frame are the features of the harmonic sinusoidal model.

Analysis

First a pitch estimation procedure is used to select a fundamental frequency for each frame. Then, for each frame, the following function, which is a type of STFT, is calculated for the harmonic frequencies:

$$S(\omega) = \sum_{n=-N/2}^{N/2} s(n) \exp(-jn\omega) \tag{2.36}$$

The amplitude and phase for each harmonic, $l\omega_0$, are the magnitude and arg of $S(l\omega_0)$, respectively.

Synthesis

Earlier synthesis processes used with this model involved attempting to match harmonics from frame to frame and using a linear function to interpolate amplitudes and a cubic function to interpolate phases. This approach was superseded by a much simpler **overlap**

add (OLA) technique which gave results that were essentially indistinguishably by human listeners [McAulay and Quatieri, 1988] [B. and Paliwal, 1995].

To synthesize using the overlap add technique, each frame, $\hat{s}(n)$ is synthesized according to the following equation:

$$\hat{s}^k(n) = \sum_{l=1}^{L^k} A_l^k cos(n\omega_l^k + \theta_l^k) \tag{2.37}$$

where the k superscripts represent the kth frame, and the A_l^k, ω_l^k, and θ_l^k values are the associated amplitudes, frequencies, and phases. The output values for samples for n ranging from the location of frame $k-1$ to frame k are given by:

$$\hat{s}[n] = w_s(n)\hat{s}^{k-1}(n) + w_s(n-T)\hat{s}^k(n-T) \tag{2.38}$$

where w_s is a window subject to the following constraint:

$$w_s(n) + w_s(n-T) = 1 \tag{2.39}$$

2.3.5 Multiband Excitation Vocoder

[Griffin and Lim, 1988] introduced the **multiband excitation vocoder** which was also based on harmonics, but used a voicing decision for each harmonic and a different excitation strategy accordingly. This approach was used in an attempt to reduce the amount of "buzziness" that typically occurs in vocoded speech due to the replacement of noise in the original speech with periodic energy.

Features

The features of the multiband excitation are a pitch period for each frame, an amplitude and a binary voicing decision for each harmonic, and a phase for each voiced harmonic.

Analysis

The analysis occurs in two steps. The first step is an **analysis by synthesis** method which attempts to estimate pitch period and spectral envelope features by considering the error between the spectrum of the original speech and the estimated synthetic spectrum. The second step makes a binary voicing decision for each harmonic based on how close the spectra of the original and synthetic speech are at that harmonic.

Synthesis

Three possible synthesis methods are mentioned in the original paper, but the third one, which is based on processing in the time domain is the one selected for implementation. For each frame, the voiced and unvoiced harmonics are synthesized separately and added together.

The voiced portion of the synthetic speech, $\hat{s}_v$ is produced by summing sinusoids for the harmonics that were judged voiced:

$$\hat{s}_v(t) = \sum_m A_m(t) cos(\theta_m(t)) \tag{2.40}$$

The amplitudes, A_m, are linearly interpolated between frames and considered 0 in unvoiced frames. The phase functions, $\theta_m(t)$, are described by the following formula:

$$\theta_m(t) = \int_0^t \omega_m(\xi) d\xi + \phi_0 \tag{2.41}$$

where ϕ_0 is an initial phase, and $\omega_m(t)$ is called a "frequency track" and is linearly interpolated between consecutive frames as follows:

$$\omega_m(t) = m\omega_0(0)\frac{S-t}{S} + m\omega_0(S)\frac{t}{S} + \Delta\omega_m \tag{2.42}$$

where S is the frame advance, and $\omega_0(0)$ and $\omega_0(S)$ are the fundamental frequencies that were estimated for the frames. There are a number of additional steps to handle various boundary cases involving transitions for voiced regions to unvoiced regions and *vice versa*.

The first step of synthesizing the unvoiced portion of the synthetic speech is to window white noise, and apply an FFT to it. Then the samples from the FFT are normalized to have magnitude 1 in each unvoiced region and the resulting transform is multiplied by a spectral envelope that is constructed by interpolating between the envelope samples from the analysis phase. The weighted overlap add method is used on the resulting transform to produce the unvoiced portion of the synthetic speech.

2.3.6 Harmonic Plus Noise Models

Stylianou [1996] describes three **Harmonic Plus Noise Models (HNM)s**. The first one, called HNM1 is the most tractable of the three and has been used in speech synthesis and voice transformation. The basic premise behind HNMs is that speech frames can be divided into two frequency bands, where the lower band consists of voiced frequencies and the upper band consists of unvoiced frequencies. The boundary between the two bands is allowed to move between frames, and can be considered to be at 0 for unvoiced frames. The voiced/unvoiced patterns that are allowable in HNMs can be seen as a subset of the possibilities for the multiband excitation model.

In a HNM, the lower band is represented by a sum of sinusoids at harmonic frequencies, and the upper band is represented by white-noise-excited LPC.

Features

For each frame, HNM features include a binary voicing decision and parameters for an LPC analysis of the frame. If the frame is voiced, they also include a fundamental frequency, a maximum voiced frequency, and amplitudes and phases for all harmonics up to the maximum voiced frequency.

Analysis

Analysis consists of a **fixed-frame** portion followed by a **pitch-synchronous** portion. The fixed-frame step estimates the pitch every 10 ms using a pitch estimation technique that is similar to the one used in the multiband excitation vocoder. The following voicing decision for each frame is performed by creating a synthetic spectrum based on harmonic sinusoids, comparing it to the original frame, and deciding based on the size of the error. The synthetic spectrum, $\hat{S}$, is created by summing sinusoids whose frequencies are integer multiples of the fundamental frequency estimate derived from the pitch estimate and whose amplitudes and phases are taken from samples from a DFT of the original signal, S.[5] The error is calculated based on the following function, which compares the synthetic spectrum to the original spectrum in a band from a little below the fundamental frequency estimate,$\hat{f}_0$, to a little above four times the fundamental frequency estimate.:

$$E = \frac{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} (|S(f)| - |\hat{S}(f)|)^2}{\int_{0.7\hat{f}_0}^{4.3\hat{f}_0} |S(f)|^2} \tag{2.43}$$

[5] In practice, using a longer frame increases the frequency resolution and typically enables finding DFT samples closer to the harmonics. Stylianou [1996] pads the frame with zeros to a length of 4096 before applying the FFT.

In practice, these values are calculated as discrete sums.[6] If the error is less than -15dB, the frame is considered voiced; otherwise it is considered unvoiced. The frequencies in neighborhoods of the harmonics are then tested according to various thresholds to determine what is the maximum voiced frequency, *i.e.* the boundary between the voiced and unvoiced frequency bands. After this is performed, the fundamental frequency estimates for the voiced frames are refined using the voiced frequencies. The refined fundamental frequency estimate for a frame is:

$$\hat{f}_0 = \arg\min_{f_i} \sum_{i=1}^{L_m} |f_i - i\hat{f}_0|^2 \tag{2.44}$$

where L_m is the number of voiced frequencies in the frame, and the f_is are the voiced frequencies.

The pitch-synchronous portion of the analysis begins by using the pitch estimates from the fixed-frame portion to determine the time points in the signal to be analyzed. They are chosen by advancing the length of the pitch period for the voiced frames and by 10ms for the unvoiced frames. It is assumed that the fundamental frequencies and maximum voiced frequencies vary little over the short term, so the pitch synchronous time analysis points use these values from the closest fixed-frame analysis points. Then the amplitudes and phases for each pitch-synchronous frame are calculated by attempting to minimize a least squares error criterion between the original speech frame and the frame that would be produced by a sum of harmonic sinusoids with those amplitudes and phases. This minimization can be performed using linear algebra, and the solution involves the inversion of a Toeplitz matrix. If the additional assumption that the interaction among the harmonics is insignificant is made, the solution can be simplified, and the amplitudes can be found using the following formula:

$$A_k = \frac{\sum_{t=t_a-N}^{t_a+N} w^2(t)s(t)e^{-j2\pi k f_0 t}}{\sum_{t=t_a-N}^{t_a+N} w^2(t)} \tag{2.45}$$

where A_k is the complex coefficient for the kth harmonic, *i.e.* the amplitude is $|A_k|$ and the phase is $arg(A_k)$; t_a is the analysis time point; N is the pitch period; $w^2(t)$ is the square of the window function; $s(t)$ is the original speech sample at time t; and f_0 is the fundamental frequency.

The rest of the pitch-synchronous analysis consists of determining parameters for the unvoiced portion of the speech. This consists of performing LPC analysis on each frame and storing normalized lattice filter coefficients along with the estimated variance for each analysis time point.

[6] It is very important to include the frequency effect of the window when creating the synthetic spectrum.

Synthesis

HNM synthesis involves producing signals for the lower frequency bands, based on harmonics, producing signals for the upper frequency bands, based on white-noise excited LPC synthesis, and adding the signals.

The basic idea for the synthesis of the harmonic part is to generate it from the following equation:

$$\hat{h}(t) = \sum_{k=0}^{L(t_s)} a_k(t_s) cos(\phi_k(t_s)) + k2\pi f_0(t_s)t) \tag{2.46}$$

where t_s is the synthesis time point, $L(t_s)$ is the associated number of harmonics, $a_k(t_s)$ is the amplitude of the kth harmonic at that time, $\phi_k(t_s)$ is the phase of the kth harmonic at that time, and $f_0(t_s)$ is the fundamental frequency at that time. In practice, synthesizing each frame in this manner and concatenating them can cause undesirable audio artifacts due to the discontinuous changes in the parameters. A few techniques for handling this problem are:

1. interpolation of the amplitudes and phases between synthesis points,
2. performing overlap-add to combine the frames, and
3. improving phase coherence by using a global center-of-gravity assumption [Stylianou].

These techniques can be used singly or in combination.

The synthesis of the noise part proceeds by generating white noise, using it as the excitation of the normalized LPC filter for the frame, and multiplying it by the variance that was stored by the frame. If the frame is voiced, a high-pass filter is then applied with a cut-off at the maximum voiced frequency for the frame, and a time-domain envelope is applied. For unvoiced frames, these additional steps are not performed. Finally, overlap-add is used to combine the noise parts from different frames.

2.4 Cepstral Analysis

Another popular representation of speech is **cepstral coefficients**. The original definition of the power cepstrum was given by Bogert et al. [1963] as the power spectrum of the logarithm of the power spectrum, though there are a number of variations [Childers et al., 1977]. A typical definition of cepstral coefficients used in speech models is the magnitude of the inverse DFT of the logarithm of the magnitude of the DFT. Cepstral parameters can also be derived from linear prediction coefficients and can be based on warped frequency scales that correspond more closely to human perception.

2.4.1 LPCC

The following recursive algorithm that derives cepstral coefficients from linear prediction coefficients [Rabiner and Juang, 1993]:

$$c_0 = \ln(\sigma^2) \tag{2.47}$$

$$c_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, 1 \leq m \leq p \tag{2.48}$$

$$c_m = \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, m > p \tag{2.49}$$

where c_m is the mth cepstral coefficient, σ^2 is the gain parameter from the LPC analysis, and a_m is the mth linear prediction coefficient. The coefficients derived by this process are called **Linear Prediction Cepstral Coefficients (LPCC)**.

2.4.2 MFCC

The **mel scale** is an experimentally derived measure of humans' perceptions of pitch that was described by Stevens and Volkmann [1940], who give a conversion formula:

$$mel = 1127.01048 \log(f/700 + 1) \tag{2.50}$$

where f is the original frequency. This scale is popular among the ones that have been used to warp frequencies during cepstral analysis. One common way of creating **Mel-Frequency Cepstral Coefficients (MFCC)** is to create a bank of triangular filters that is evenly spaced along the mel scale, and to use the output powers of the filters on speech input as follows [Rabiner and Juang, 1993]:

$$\tilde{c}_n = \sum_{k=1}^{K} (log \tilde{S}_k) cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right] \tag{2.51}$$

where c_n is the nth MFCC, and $\hat{S}_k$ is the power from applying the kth filter to the speech signal.

2.4.3 Mel Log Spectral Approximation Filter

The **Mel Log Spectral Approximation Filter (MLSA)** was introduced by Imai [1983]. This filter has been used in Hidden Markov Model-based speech synthesis [Tokuda et al., 2000], and is also used in the baseline voice transformation system used in this thesis. The parameters from the MLSA are based on a frequency scale that is warped by an all-pass filter and approximates the mel scale. The analysis algorithm derives a "true" envelope by adjusting the MLSA parameters so their derived spectrum goes through the peaks of the original spectrum. This is supposed to make the MLSA parameters more closely match the spectral envelope of the original speech than the traditional MFCCs do.

2.5 Summary

When automatically processing speech for tasks, such as voice transformation, it is necessary to have a convenient, tractable representation of speech. The representation of the speech is a speech model, which consists of features, and the methods of producing the features from the speech and producing the speech from the features are called analysis and synthesis, respectively.

Three basic techniques that have demonstrated convenience and tractability for creating speech models for a wide range of speech applications are linear prediction, Fourier analysis, and cepstral analysis. In practice, these techniques can be used in combination, so they shouldn't be considered distinct categories of speech models, but distinct tools that can be used to build them. It is, however, possible to trace the development of various speech models and see how they refine the use of these techniques.

Three speech models that will be of particular interest in the following chapter on voice transformation are LSF, HNM, and MLSA. The current chapter described the building blocks for these models. LSF is based on a transformation from the linear prediction features. HNM comes from a family of techniques originally based more on methods inspired by Fourier analysis and eventually came to include some linear prediction techniques. MLSA is a form of cepstral analysis. Cepstral analysis, itself, is based on spectral analysis, which is typically performed using Fourier analysis or linear prediction.

Chapter 3

Voice Transformation

3.1 Introduction

In this document, **voice transformation**, also known as **voice conversion** and **voice morphing**, is defined as the process of making speech from one speaker sound as if it had been spoken by another. These speakers are called the **source speaker** and **target speaker**, respectively.

Voice transformation is important for both scientific and practical reasons. On a fundamental level, the study of mappings of speech from one person to another is the study of how speech differs among people. Implicit in this investigation is the consideration of which parts of the speech signal reflect general features shared among speakers for a particular language (and in some cases, even across languages) and which parts reflect specific features that relate to an individual's identity. From the practical perspective, mappings from one speaker to another enable or simplify numerous applications, some of which will be discussed in Section 3.2.

The task of transforming speech from one identity to another is difficult and the best way to pursue this goal is still an open question. In a general sense, speech from two people can vary in many ways, including **word choice**, **prosody**, **phonetics**, and **acoustics**. Word choice differs among speakers when they use alternate words to convey the same or similar meanings. Prosody differs among speakers who speak at different pitches and volumes. It also differs when speakers speak at different rates and use different relative durations for various speech sounds. Phonetics differ in speech when people pronounce the same word using different phonemes. Acoustics differ among speakers who have vocal tracts of different sizes and shapes that change the audible waveforms they produce when they speak.

Due to the difficulty of the general problem of voice transformation, it is typical to limit the scope by focusing on certain levels. Current voice transformation techniques primarily address the acoustic level, and to a lesser extent, the prosodic level, although there have been occasional attempts to more fully model the prosodic level and to incorporate the phonetic level into models [Abe, 1991]. The typical focus on the acoustic level with a minimal use of prosodic features owes to the strengths of current speech models, which are usually source-filter models. In this context, the source includes pitch and power information, which would most closely correspond to the prosodic level in a linguistic sense, and the filter includes spectral information, which would be most closely related to the acoustic level in a linguistic sense. Furthermore, whatever features that are chosen for modeling speakers are also subject to the constraint that inter-speaker mappings for these features must be feasible. A discussion of some features and models that have been used for voice transformation by various researchers, with a view towards the techniques used in our work, is in Section 3.3. More specific details about the baseline voice transformation technique used in the experiments in this document is in Section 3.4.

One common practice that simplifies the problem of voice transformation by limiting its scope is to have the source and target speakers read the same text, though there are systems which do not depend on this [Sündermann et al., 2006]. When the speakers read the same material, differences in word choice are not even considered and further differences that might manifest themselves in spontaneous, conversational speech are also ignored. Another way of limiting the scope is to reduce prosodic differences by having target speakers attempt to mimic source speakers [Kain, 2001].

Regardless of the scope of the transformation, there is always the question of how good a voice transformation actually is. Numerous ways of evaluating voice transformation have been devised. This is the topic of Section 3.5.

3.2 Applications

Voice transformation has many applications. One that is frequently mentioned in the literature is the ability to more easily build synthetic voices with new identities and styles [Kain, 2001]. Without voice transformation, the typical approach to building a concatenative speech synthesizer involves recording a person reading thousands of sentences. This size is necessary to cover the majority of phonetic cases that would be necessary for new synthetic utterances. In addition to difficulties from the time and effort involved in reading that many sentences, most people are not able to read that quantity in a consistent manner, and inconsistencies in speakers' deliveries can decrease quality in the final synthesizers. With the standard approach to concatenative speech synthesis, the construction of every new voice requires a person to read the entire corpus of thousands of sentences. Another approach to the problem of creating a synthetic voice with a new identity is to take an existing synthesizer and apply voice transformation to its output so it sounds like a new speaker. This approach is appealing because voice transformation training requires recording far fewer sentences. One estimate is that recording 50 sentences per speaker is sufficient for GMM-based voice transformation [Mesbahi et al., 2007]. Using a corpus of this size greatly reduces the cost of building a new synthetic voice and lessens the burden on speakers who will have an easier time being consistent.

Another application of voice transformation is to try to defeat **speaker identification (SID)** systems [Pellom and Hansen, 1999] [Masuko et al., 2000] [Jin et al., 2008]. SID systems take input speech and attempt to determine whether it was spoken by a particular person. Such systems have security applications. Voice transformation can be used to impersonate a speaker in an attempt to fool an SID system. Another related application involves privacy. Voice transformation techniques can be used to **de-identify** speech by making it difficult for people and SID systems to determine who is speaking [Jin et al., 2009].

In his dissertation, Kain [2001] lists a number of additional applications. One is to provide an identity to speech coded at such a low bandwidth that its identity has been removed [Schmidt-Nielsen and Brock, 1996]. Another is to apply a speaker's identity to speech in an unknown foreign language, for example in speech-to-speech machine translation or movie-dubbing [Abe et al., 1991] [Abe et al., 1990b]. Yet another is to improve the intelligibility of acoustically impaired speech [Abe et al., 1991] [Mizuno and Abe] [Stylianou et al., 1998] [Kain et al., 2004]. A more recent effort in this last area includes the transformation of **Non-Audible Murmur (NAM)** speech to regular speech [Toda and Shikano, 2005]. Such technology could allow people to have public conversations and yet be nearly unheard by others.

3.3 Related Voice Transformation Work

The amount of literature on Voice Transformation is sizable and goes back to at least 1985 [Childers et al., 1985]. One of the main lines of research has been the investigation and subsequent refinement of linear mappings between clusters of source and target speaker spectral features. Earlier work in this area used **Vector Quantization (VQ)** codebook-based techniques to create hard clusters of spectral features [Abe et al., 1990a]. These techniques were superseded by **Gaussian Mixture Model (GMM)** mapping-based techniques which created soft clusters of spectral features [Stylianou et al., 1995a]. Originally, a GMM was learned for the source speaker data only, but a later refinement was to learn a GMM on the joint space of features from both speakers [Kain, 2001]. Later work tried to compensate for the excessive smoothing of the GMM mapping-based technique by creating a hybrid approach with **Dynamic Frequency Warping (DFW)** [Toda, 2003]. More recent developments include using maximum likelihood estimation to find model parameters, modeling dynamic features to improve estimated spectral feature trajectories, and incorporating **global variance** into the models to again compensate for the excessive smoothing of the original model [Toda et al., 2007].

3.3.1 VQ Codebook-Based Techniques

Abe *et al.*

A Voice Transformation technique based on VQ codebooks was introduced by Abe et al. [1988]. The process for learning a mapping between the codebooks for two different speakers went as follows:

1. Extract LPC coefficients from frames from words recorded by source and target speakers.
2. Use VQ on the resulting LPC coefficient vectors.
3. Align each word between speakers using Dynamic Time Warping (DTW).
4. Create histograms for the vector correspondence between speakers.
5. Create a mapping codebook that uses the histograms to create linear combinations of the target speaker's vectors.
6. Repeat the DTW, histogram creation, and mapping codebook creation until the mapping codebook is considered to be good enough.

Mapping codebooks were also created for pitch and power, but the procedures were slightly different as the values were scalars, and the maximum histogram occurrence was used in creating the pitch mapping codebook.

Transformation was performed by extracting the LPC coefficients, pitch, and power from a new recording by the source speaker, decoding them using the previously learned mapping codebooks, and synthesizing from the resulting values.

In the analysis of their experimental results, the authors concluded that both pitch and spectrum were necessary for the individuality of the speech, and that neither alone was sufficient for good voice transformation. All of the male-to-female transformed speech was judged female by listeners, and 65% of the male-to-male transformed speech was judged more similar to the target speaker than the source speaker by listeners. Although this represented some level of success, there was still much room for improvement. One of the problems with the original codebook based technique was that it led to a mapping that had many step-like discontinuities, which degraded the quality of the transformed speech.

3.3.2 GMM Mapping-Based Techniques

A few techniques were tried to smooth out the discontinuities inherent in the original codebook mapping technique. The most prevalent one is to replace the codebook map with a Gaussian Mixture Model (GMM) map. The underlying idea is that a GMM, which is continuous probability distribution that is described in more detail below, can be used to model the feature vectors from the source speaker or both speakers. Various algorithms can then use this GMM to perform voice transformation.

In order to understand a Gaussian Mixture Model, it is necessary to first understand the multi-dimensional Gaussian distribution. A p-dimensional Gaussian distribution, $\mathcal{N}$, is a continuous probability distribution over $\mathcal{R}^n$ that is parametrized by a p-dimensional mean vector, μ, and a p-by-p covariance matrix, Σ. The probability density function of a p-dimensional Gaussian distribution for $x \in \mathcal{R}^n$ is:

$$\mathcal{N}(x; \mu, \Sigma) = \frac{|\Sigma|^{-1/2}}{(2\pi)^{p/2}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

A GMM is a weighted sum of Gaussian probability density functions subject to constraints and defines a probability distribution function, $p(x)$, for a vector, $x \in \mathcal{R}^n$, as follows:

$$p(x) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

where M is the number of Gaussian components, μ_i is the mean of the ith Gaussian, Σ_i is the covariance matrix of the ith Gaussian, and the α_i weights are subject to the constraints that

$$\forall i, \alpha_i \geq 0$$

$$\sum_{i=1}^{M} \alpha_i = 1$$

The parameters of a GMM are typically learned by taking a number of data vectors that are assumed to have been generated by the GMM and using the EM algorithm [Dempster et al., 1977] to estimate parameters that will maximize the likelihood of the data vectors. In the following approaches, these data vectors consist of features derived both from source and target speakers.

Different researchers have devised various ways to use GMMs to perform transformations from source speakers to target speakers. Such methods are described below in more detail.

Yannis Stylianou *et al.*

The approach of using GMMs for Voice Transformation appears to have originated with Yannis Stylianou and his coauthors [Stylianou et al., 1995a] [Stylianou and Cappé, 1998]. In this work, acoustic features were derived from a **Harmonic + Noise Model (HNM)** [Stylianou et al., 1995b]. This model divides speech into a low band modeled by harmonically-related sine waves, and a high band modeled by noise modulated by a time-domain amplitude envelope. This representation was chosen because modifications to duration and pitch seemed to be relatively straightforward to perform and of high-quality. For voice transformation, the HNM was simplified a little to make processing more convenient. The maximum voiced frequency, which is the cutoff between the frequency bands, is allowed to vary with time in the original HNM, but in the version used with voice transformation, it was fixed at 4 kHz voiced frames. Also the analysis was changed so HNM parameters were collected at a fixed rate of 10ms between analysis points, which is different from the pitch-synchronous approach used in the original HNM.

Features were extracted from the low band by converting the harmonic frequencies to a Bark frequency scale and then extracting cepstral coefficients using a regularization technique [Cappé et al., 1995]. The 1st through 20th coefficients from the source and target speaker speech were then aligned using **Dynamic Time Warping (DTW)**.

The low and high bands were converted separately, and the cutoff was fixed at 4 kHz. The low band was converted using a GMM mapping, and the high band was converted using two different filters. One filter was for voiced frames, and the other was for unvoiced frames.

The transformation function was not based on a proper statistical model, but was constructed by analogy to the solution of the minimum mean square estimation function for the case where a single Gaussian distribution was used to model the features for each speaker. For a source speaker feature vector, x, the transformation function had the form:

$$\mathcal{F}(x) = \sum_{i=1}^{M} P(C_i|x)[\nu_i + \Gamma_i \Sigma_i^{-1}(x - \mu_i)] \tag{3.1}$$

where $P(C_i|x)$, Σ, and μ were taken from a GMM that had been previously learned on the source speaker's training data. Σ was the covariance matrix, μ was the mean, and $P(C_i|x)$ was the conditional probability density of the ith Gaussian class given data vector x:

$$P(C_i|x) = \frac{\alpha_i \mathcal{N}(x; \mu_i, \Sigma_i)}{\sum_{j=1}^{M} \alpha_j \mathcal{N}(x; \mu_j, \Sigma_j)} \tag{3.2}$$

where a_i is the weight of the ith class.

The ν_i and Γ_i values were determined from the data by minimizing the mean squared error of the transformed vectors with respect to the target speaker feature vectors on the training set:

$$\{\hat{\nu_i}, \hat{\Gamma}_i\} = \arg \min_{\{\nu_i, \Gamma_i\}} \sum_t ||y_t - \mathcal{F}(x_t)||^2 \tag{3.3}$$

where t is an index vector over the alignments of source and feature vectors, and x_t is a source speaker feature vector aligned with a target speaker feature vector, y_t.

In the single Gaussian case this was based on, there would have only been one class, so the $P(C_1|x)$ factor would have been unnecessary as it would have always equaled 1, and a minimum mean squared error solution based on a proper statistical model would have set ν_1 to the target speaker's feature vector mean, and Γ_1 to the cross-covariance matrix of the aligned source and target vectors: $\Gamma_1 = E[(y - \nu_1)(x - \mu_1)^T]$. In the multiple Gaussian case that was used for voice transformation, the ν_i and Γ_i values were determined by solving a more complicated system of linear equations and do not necessarily correspond to means and cross-covariances.

A listening evaluation was performed with 20 people. One test was an "XAB" test (called an "ABX" test in other literature [Kain, 2001]) which provided an example "X" of converted speech and asked listeners to compare it with examples "A" and "B" of source and target speech and decide which was more similar. Listeners overwhelmingly found the voice conversion technique with 16 Gaussians in the mixture model to perform better than a technique which only modified prosody, and further improvements occurred when 64 Gaussians were used and the same sentence was used for all three examples. In this last example, 97% of the converted utterances were considered to be closer to the target speech. This demonstrated that converting spectral characteristics in addition to prosodic characteristics improves the quality of Voice Transformation in terms of speaker identity. Another test called an “opinion” test (called a pair comparison test by Kain [2001]) involved presenting listeners with pairs of utterances where they could be actual speech, converted speech, or prosodically-modified speech. The listeners were then asked to rate the similarities of the pairs on a scale from 0 meaning “identical” to 9 meaning “very different”. The results of this test corroborated the results from the “XAB” test.

Alex Kain’s Ph.D. Dissertation

Alex Kain’s Ph.D. dissertation [Kain, 2001] extended the idea of using GMM mappings in Voice Transformation in a few more directions. He tried different acoustic features, used a different alignment strategy, and also attempted to predict residuals based on his model.

In this work, speech was parametrized in a different way. Instead of spacing frames every 10ms, they were spaced pitch synchronously. Unvoiced speech was assumed to have a constant pitch of 125 Hz. Speech was represented using a harmonic sinusoidal model. Then, the frequencies were warped according to the Bark scale, and Linear Predictive Coding (LPC) coefficients were calculated. Because LPC coefficients are difficult to modify in a stable manner, Line Spectral Frequencies (LSF) [Itakura, 1975] were then computed from the LPC coefficients. These Line Spectral Frequencies were used as features in the Gaussian Mixture Model.

Alignment of source and target features proceeded differently from Stylianou’s DTW approach. “Time marks” were created by performing HMM-based forced alignment on the source and target speech using phonetic transcriptions. The HMM states were aligned, and frames were repeated in or deleted from the target speech to match the length of the source speech.

Although the transformation function, like Stylianou’s, was based on a GMM, the transformation function used the GMM differently. Instead of only being trained on the source speaker’s feature vectors, the GMM was trained on joint vectors that included features from both the source and target speakers from aligned frames. This meant that the GMM classes were based on both speakers and not just the source speaker. A different transformation function was also used. For a source speaker feature vector, x, the transformed feature vector was:

$$\mathcal{F}(x) = \sum_{i=1}^{M} P(C_i|x)[\mu_{y,i} + \Sigma_{yx,i}\Sigma_{xx,i}^{-1}(x - \mu_{x,i})] \tag{3.4}$$

where $\mu_{y,i}$ was the mean of the target speaker feature vectors in the ith class, $\Sigma_{yx,i}$ was the lower left quadrant of the joint GMM covariance matrix representing the covariance between the target and source speaker features, $\Sigma_{xx,i}$ was the upper left quadrant of the joint GMM covariance matrix representing the covariance of the source speaker features, and $\mu_{x,i}$ was the mean of the source speaker feature vectors.

This transformation function differed from Stylianou's in that all terms were derived from training a GMM on the joint feature space, and no additional system of linear equations had to be solved after the GMM training. It did, however require training a GMM on vectors with twice the number of dimensions as were in a single speaker's feature vectors. Also, synthesis proceeded differently due to differences in the speech models and frame selection.

One of this dissertation's major contributions was the idea of attempting to predict the LPC residual for the target speaker. This was performed by deriving cepstral coefficients from the LPC coefficients and training a GMM classifier for deciding class membership for vectors of such cepstral coefficients. Classifications from the GMM were used to create a codebook of residuals, where residuals were represented as 100-point samples. In order to predict a residual for a vector of LPC cepstral coefficients, the GMM parameters and residual codebook were used to produce a weighted sum of residual codebook vectors, which was the estimate. Residual prediction was found to improve the speaker identity of transformed speech in subjective evaluations.

Tomoki Toda's Ph.D. Thesis

Part of Tomoki Toda's Ph.D. Thesis [Toda, 2003] was concerned with using GMM mapping techniques with Voice Transformation. He created a Voice Transformation technique that used a combination of GMM mapping and Dynamic Frequency Warping (DFW). Subjective evaluations supported the conclusion that the quality of the synthesized speech was improved while the speaker identity remained just as good, when compared to a conventional GMM mapping technique that did not use DFW.

The acoustic features, analysis, and synthesis in this thesis differed from the ones in the other approaches. For the acoustic features, STRAIGHT [Kawahara et al., 1999] analysis was used to produce a smoothed spectrum, and MFCCs were calculated from this spectrum at a fixed interval. Alignment of source and feature vectors was performed by removing silence frames and then using a Dynamic Time Warping algorithm on the remaining frames. Then an iterative process was used to improve the alignment. First, a GMM mapping was trained on the source and target speakers' features. Then alignment was performed again using the transformed version of the source speaker's features and the target speaker's features. This process was repeated iteratively until the change in the

Mel-Cepstral Distortion was less than a threshold.

Toda *et al.*

Later innovations by Tomoki Toda and coauthors included using the EM [Dempster et al., 1977] algorithm to produce maximum likelihood estimates of the target features instead of just using expectations, using additional "dynamic features" (weighted windows of features), and changing the optimization function from maximum likelihood to a weighted mixture of maximum likelihood and global variance [Toda et al., 2007].

One of the problems with the previous approaches to voice transformation was that they were only focused on predicting features for one frame at a time. They did not attempt to model the frame-to-frame dependencies of features. Toda and coauthors created a GMM mapping based approach based that included **dynamic features** in the estimation procedure. These dynamic features were created by weighting the feature values of a small number of frames around the current frame and summing them.

Although a GMM was still used in the creation of a transformation function that was used with these additional dynamic features, a new technique for constructing the transformation function was created. Instead of focusing on minimizing mean squared error, the objective was to maximize the likelihood of the target speaker's feature vectors. The attempted to estimate a sequence of target speaker vectors, $\hat{\mathbf{y}}$, according to the following formula:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{Y}|\mathbf{X}, \lambda) \tag{3.5}$$

where $\mathbf{y}$ was a sequence of target speaker feature vectors, $\mathbf{Y}$ was the corresponding sequence of target speaker feature vectors augmented with their dynamic features, $\mathbf{X}$ was the sequence of source speaker feature vectors augmented with their dynamic features for the aligned frames, and λ was the set of GMM parameters. The solution to this equation was estimated by assuming the conditional probability took the form of a GMM, using the EM algorithm to search for a local maximum, and using linear algebra to relate $\mathbf{y}$ to $\mathbf{Y}$. The conversion procedure based on this transformation function was called **Maximum Likelihood Parameter Generation (MLPG)** [Toda et al., 2007].

An approximate solution that was more efficient to calculate was also created by changing the procedure to first estimate and fix the sequence of mixture components. It was found that this did not lead to a large difference in the accuracy of the transformation function.

Another problem with previous voice transformation techniques was that they tended to smooth out too much of the variation in speech. For this reason, Toda et al. [2007]

created a regularization strategy based on including a **Global Variance (GV)** term in the function to be optimized. An additional term was added to the likelihood function from the MLPG approach, which was a probability based on the sequence of sample variances of the target speaker feature vectors. Furthermore, an extra parameter was added to enable relative weighting between the original likelihood factor, and the new global variance factor. The objective function became:

$$P(\mathbf{Y}|\mathbf{X},\lambda)^{\omega}P(\mathbf{v}|\kappa) \tag{3.6}$$

where $\mathbf{Y}$, $\mathbf{X}$, and λ have the same meanings as in the MLPG case, ω controls the relative weight of the likelihood and the global variance probabilities, $\mathbf{v}$ is the sequence of variances of the target speaker feature vectors, and κ is a set of parameters for a single Gaussian, which is used to model the variances. Another procedure involving EM was created to use this new objective function to transform feature vectors. In addition, a more efficient version using an approximate solution based on estimating and fixing the sequence of mixture components. Again, it was found that this did not affect performance significantly.

It was found that using global variance in these led to improved quality of transformed speech based on human listening tests, but worsened its score according to a calculated measure.

One thing to note is that the choice of a Gaussian to model for the variances seems a bit unusual as variances must be non-negative, and have Chi-square distributions.

3.4 Festvox Voice Transformation Scripts

The FestVox distribution [Black and Lenzo, 2000] includes scripts written by Tomoki Toda that perform voice transformation. They are described here in more detail because the voice transformation experiments in this thesis were based on modified versions of these scripts.

3.4.1 Training Process

The baseline voice transformation training process is depicted in Figure 3.4.1 and described below.

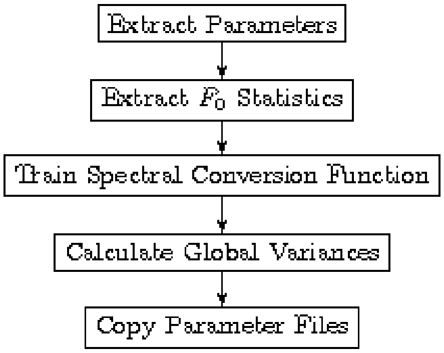


Figure 3.1: Baseline Voice Transformation Training Process

Extract Parameters

For all of the source speaker and target speaker audio files in the training set, produce estimates of the fundamental frequency (F_0) on 25ms frames with an advance of 5ms. If a frame is judged unvoiced, record a 0 for its fundamental frequency.

For all of the source speaker and target speaker audio files in the training set, extract 24 **MCEP**s, which are the MLSA parameters which approximate MFCCs, per frame. Again, the framesize is 25ms and there is an advance of 5ms between frames.

Calculate F_0 Statistics

Take the logarithms of the non-zero F_0 values for the source and target speaker training utterances, and record the means and standard deviations for each speaker.

Train Spectral Conversion Function

Of the steps in the overall voice transformation process, the most complicated is the training of what is called the spectral conversion function in the FestVox scripts. This is the GMM that is used to map the filter features from the source speaker to the target speaker. By default these features are MCEPs and features derived from them. This map does not account for prosodic features such as pitch, power, and duration. These are either handled separately (pitch) or implicitly (power and duration).

Dynamic features are features that are derived from a combination of MCEP vectors sampled at different times. They are used to model changes in MCEPs over time. The

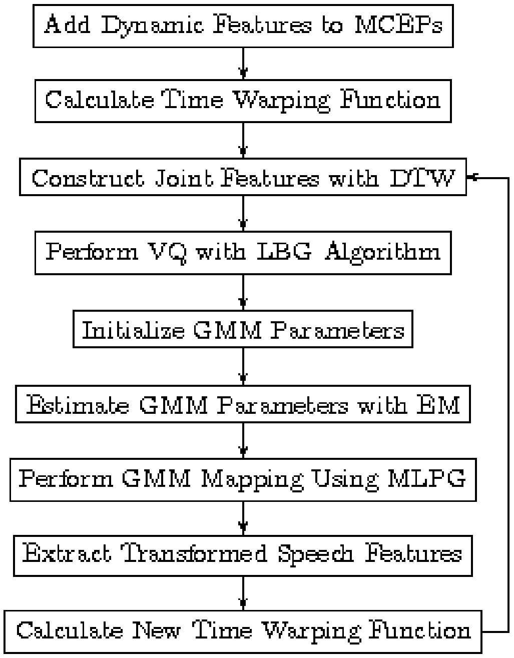

Add Dynamic Features to MCEPs
Calculate Time Warping Function
Construct Joint Features with DTW
Perform VQ with LBG Algorithm
Initialize GMM Parameters
Estimate GMM Parameters with EM
Perform GMM Mapping Using MLPG
Extract Transformed Speech Features
Calculate New Time Warping Function

Figure 3.2: Baseline Voice Transformation Training Process

baseline formulation of dynamic features is that they are a linear combination of an MCEP vector with the previous (sampled 5ms earlier) and following (sampled 5ms later) MCEP vector with the coefficients specified in the following formula:

$$-0.5M_{t_{f-1}} + M_{t_f} + 0.5M_{t_{f+1}} \tag{3.7}$$

where M_{t_f} denotes the MCEP vector at the time of frame f.

Different people speak at different rates. Furthermore, the time spent on corresponding sounds in utterances differs from speaker to speaker. A **time warping function** is used to match parts of one speaker's utterance with another speaker's utterance. It is a mapping from times in one utterance to times in another. This, of course, leads to the question of how the map is derived in the first place. First, it is assumed that there is a reasonable metric for comparing the similarity of MCEP vectors from the different speakers. In this case, **Mel-Cepstral Distortion** (MCD) between a source speaker feature vector, x, and a target speaker feature vector, y, is used:

$$MCD(x,y) = \frac{10}{\ln 10}\sqrt{2\sum_{i=1}^{N}(x_i - y_i)^2} \tag{3.8}$$

where N is the number of dimensions in the feature vector, m_{s_i} is the ith value in a source speaker's MCEP vector, and m_{t_i} is the ith value in a target speaker's MCEP. In this step, the feature vector for each frame consists of 24 MCEP values plus 24 corresponding dynamic features for a total of 48 dimensions.

Once there is an adequate measure, there is still the question of how this relates to finding the best path through all the vectors. The naive consideration of all paths separately would lead to a search through a number of possibilities that is exponential in the length of an utterance. Fortunately, there is an efficient method to find the best path called **Dynamic Time Warping** (DTW) [Itakura, 1975]. The use of DTW in the baseline voice transformation scripts is based on the assumption that the closeness of a path of vectors from the source speaker to a path of vectors from a target speaker can be determined by considering only the combination of the distance between the source and target vectors with the scores for a few paths that go through previous time samples.

The specific rule used in the baseline voice transformation scripts for determining the score of a path through source and target speakers is defined recursively such that at time t, the score is

$$\begin{aligned}
\min(\quad & MCD(x_{t-2}, y_{t-1}) + MCD(x_{t-1}, y_t) + MCD(x_t, y_t), && (3.9)\\
& MCD(x_{t-1}, y_{t-2}) + MCD(x_t, y_{t-1}) + MCD(x_t, y_t), && (3.10)\\
& MCD(x_{t-1}, y_{t-1}) + 2MCD(x_t, y_t)) && (3.11)
\end{aligned}$$

The DTW rule is applied according to the following procedure:

1. Create a matrix that has the width of the x utterance and the height of the y utterance and initialize all values to infinity. The columns correspond to frames from the x utterance, with the first frame on the left. The rows correspond to frames from the y utterance, with the first frame on the bottom.

2. Calculate the MCD between the initial frames, x_0 and y_0, of each utterance and record that value in the corresponding lower-left matrix entry.

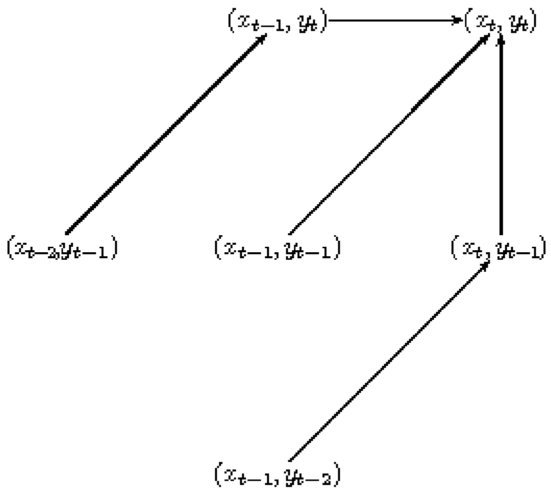


Figure 3.3: Rule Used for Dynamic Time Warping

3. Consider the next matrix column to the right, which corresponds to the next frame in the x utterance. For each entry in this column that can be reached by an application of the DTW rule, in the sense that a finite minimum value can be calculated based on entries in columns to the left, record the minimum value for the rule.

4. Repeat the previous step through the rightmost matrix column.

If the upper-right corner of the matrix is reached, the DTW successfully aligned the two utterances. It should be noted that use of our particular DTW rule will fail in cases where one utterance is more than twice as long as the other. If the DTW succeeded, the alignment between the utterances is found by using the minimum scores to reconstruct the path that led from the lower-left corner of the matrix to the upper-right corner.

Figure 3.4.1 depicts the various paths through the matrix that are considered by the DTW rule in the determination of the value at position (x_t, y_t).

The following steps in the process to learn a spectral conversion function are iterated three times by default.

The time warping function is used to create vectors from the joint feature space of the source and target speakers. Corresponding 48-dimensional feature vectors from the source and target speakers are combined to create 96-dimensional feature vectors. These consist of 24 source speaker MCEPs, followed by 24 source speaker dynamic features, followed by 24 target speaker MCEPs, followed by 24 target speaker dynamic features.

Vector quantization is performed on all of these joint feature vectors using the LBG algorithm [Linde et al., 1980] to create codebooks, which will be used to provide initial parameters for the GMM during training.

At this point, a GMM is trained using the EM algorithm. The default is to use 64 Gaussian components in the mixture.

After the training process for the GMM has converged, **Maximum Likelihood Parameter Generation (MLPG)** is used to predict target speaker MCEPs from the source speaker MCEPs (temporarily ignoring the actual target speaker MCEPs that are included in the training data). Dynamic features are generated from the predicted target speaker MCEPs, a new time warping function is constructed by applying the same DTW rule between the source target features and the predicted target speaker features, and this function is used to align source speaker features with actual target speaker features in the next round of iteration.

The GMM parameters from the third iteration are stored for use during voice transformation.

Calculate Global Variance Statistics

For each target speaker training utterance, calculate the variance of the MCEPs. Then calculate and store the mean and variance of these variances.

Copy Parameter Files

Collect the locations of the various mapping and statistics files that were created during the training process and record them in a standard place so they can be accessed easily during testing.

3.4.2 Test Voice Transformation

The previous steps in this process were all concerned with training. After the training procedure estimates parameters, they can be used to transform new utterances from the source speaker so they hopefully sound like they were spoken by the target speaker. The following procedure performs this voice transformation.

First, F_0 estimates for the source speaker utterance are made every 5ms. These estimates are **z-score mapped** based on the $\log F_0$ means and standard deviations to produce estimates for the target speaker. The z-score map from a source value, x_s, to a target value,

x_t, is defined by the formula

$$x_t = (x_s - \overline{x_s})\frac{\sigma_{x_t}}{\sigma_{x_s}} + \overline{x_t} \tag{3.12}$$

where $\overline{x}$ is the mean of the values of some variable, x, and σ_x is the standard deviation of the values of some variable, x. During training, the means and standard deviations were calculated over the $\log F_0$ estimates, so there is a small modification to the z-scoring procedure. The $\log$s of the source speaker F_0 estimates are taken, the z-score mapping computes estimates of the $\log$s of the target F_0 values, and these are converted to the target speaker F_0 estimates through exponentiation.

Next, the 0th through 24th MCEPs are extracted from the source speaker utterance every 5ms. The 0th order MCEPs from the source speaker will be used unchanged in the transformed speech to preserve the average power. Dynamic features are extracted from the 1st to 24th MCEPs, and the combination of the two is used with the GMM-based spectral conversion map to predict and MLPG using Global Variance to predict MCEPs for the target speaker utterance. The 0th order MCEPs from the source speaker are combined with the predicted 1st through 24th MCEPs for the target speaker, and the MLSA filter uses the target speaker F_0 estimates with them to produce a synthetic speech waveform.

3.5 Evaluation

3.5.1 Qualities

What would be considered successful voice transformation? We posit that in order to be successful, voice transformation must be good in terms of **naturalness**, **intelligibility**, and **identity**. **Naturalness** is how human the produced speech sound. **Intelligibility** is how possible it is to correctly understand the words that were said, and **identity** is the recognizability of the individuality of the speech.

3.5.2 Measures

Different methods have been devised to measure naturalness, intelligibility, and identity. Some are **objective** measures, which can automatically be computed from audio data. Others are **subjective** measures, which are based on the opinions expressed by humans in listening evaluations or on other human behavior.

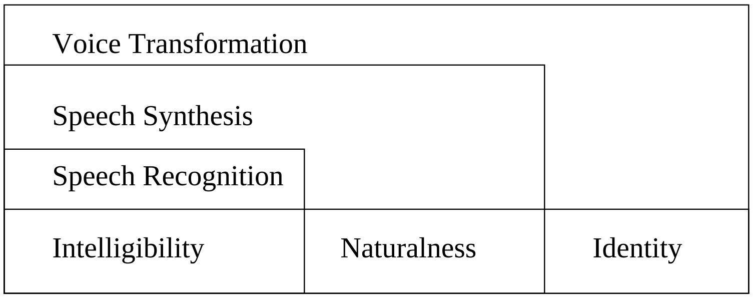


Figure 3.4: Goals of Various Speech Applications

Objective Measures

Objective measures are ones that can be automatically computed from data. Their advantage is that they are typically faster and cheaper to compute as they don't involve human experiments. Their disadvantage is that the ones we have do not directly mimic human perception, which is typically[1] the standard for judging qualities such as naturalness, intelligibility, and identity. Fortunately, the previously mentioned objective measure called mel-cepstral distortion does appear to correlate with human perception of the quality of transformed speech, though it does not match it perfectly.

Subjective Measures

Subjective measures are based on collecting human opinions and analyzing them. Their advantage is that they are directly related to human perception, which is typically the standard for judging the quality of transformed speech. Their disadvantages are that they are time-consuming, expensive, and difficult to interpret. The opinions of numerous people must be collected to evaluate a subjective measure, because opinions vary among people.

Two popular identity tests are ABX and pair comparison tests. In ABX tests, listeners are asked whether transformed utterances sound more like source speaker or target speaker utterances. In pair comparison tests, listeners are given pairs of utterances, and asked to rank their similarities on a numeric scale. Some consider the basic type of ABX test, where listeners are only asked to specify whether example "A" or example "B" is closer to example "X", inferior to pair comparison tests because it does not account for the possi-

[1] In some cases, human perception may not be the standard, because people may exhibit fallibility while performing a task with an external standard of correctness. For example, a person may be asked to identify which speakers spoke certain utterances, but limitations in the person's ability to distinguish speakers or knowledge of the specific speakers might prevent the person from correctly identifying the speakers.

bility that the transformed utterance may not sound like it was produced by either speaker [Kain, 2001]. However, pair comparison tests are more difficult to interpret because the information from the various pairs must be combined into one coherent whole that can be interpreted. Two techniques for this are Multi-Dimensional Scaling [Abe et al., 1988] and Transformation Triangle Diagrams [Toth and Black, 2006].

3.6 Summary

Voice transformation, the process of making speech from one person sound like it came from another, is an area of speech synthesis that has scientific implications and has numerous practical applications. One prominent line of voice transformation research is based on using speech models, as described in the previous chapter, to produce features for source and target speakers and to use statistical techniques to derive mappings from the source speaker features to the target speaker features. This line of work can be traced back to the codebook-based techniques of Abe through the GMM mapping technique of Stylianou, which was further refined by Kain and Toda. Our baseline system is based on the work of Toda, and is described in this chapter.

Chapter 4

Using Articulatory Position Data with Voice Transformation

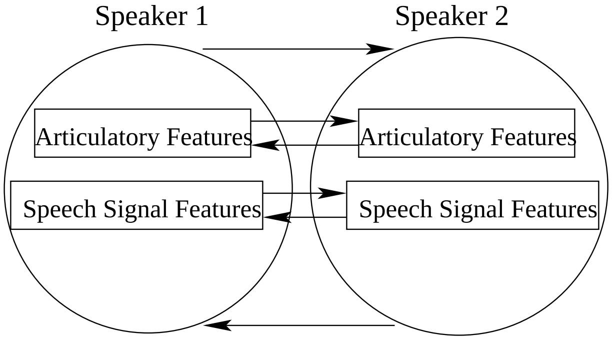
4.1 Introduction

One of the main goals of this document, as mentioned in the introductory chapter, is the investigation of the use of articulatory position data to improve voice transformation. In particular, this chapter presents the use of such data to modify the baseline voice transformation system from the previous chapter.

Articulatory position data is information on the location of articulators during speech. [1] As articulatory position data provides direct information on the physical production of speech, there is hope that it can be used to improve models for speech. In many cases, current speech models are based on features derived from the audio signal through signal processing techniques such as LPC, cepstra, or mel-cepstral coefficients. Such features are arguably either more related to the perception of speech than the production of speech or represent an attempt to indirectly reconstruct information about production. Articulatory position data is exciting in that it gives direct information about production, but it is not without its limitations. One difficulty is that it may not fully represent the important parts of production. The **Electro-Magnetic Articulograph (EMA)** data, which is used as the articulatory position data in the following experiments, consists of recordings of the positions of seven articulators in the midsagittal plane. Seven points in a plane may not be sufficient to represent lateral effects, constrictions in the vocal tract, or the shape of the tongue. Information about pitch and power will not be directly represented. However,

[1] This chapter is adapted from a workshop paper [Toth and Black, 2007].

Figure 4.1: *Transformation of Articulatory and Speech Data*



there may still be usable information even though the information is not complete, and there is evidence, at least for speech recognition, that it can help [Wrench and Richmond, 2000] [Uraga and Hain, 2006].

Another difficulty is that articulatory position data is hard to collect and this makes it fairly rare. In most cases, audio recordings of speech made by microphones are not accompanied by corresponding articulatory position data. Thus, there is the additional question of whether a limited amount of articulatory position data, which was collected for only a few speakers, can be used with audio recordings from speakers from whom articulatory position data was not collected. There has been some work in this area as well [Toth, 2005], which will be discussed in the following chapter. In this context, it is natural to ask whether using articulatory position data can provide useful modeling information beyond what is available from the audio signal and for what tasks is it helpful.

The following experiments attempt to extend the use of articulatory position data to voice transformation. A high-level view of the approach taken in this chapter can be seen in Figure 4.1. The general idea is that, in addition to mapping features derived from the speech signal data from one speaker to another, we can also map features derived from articulatory data from one speaker to another. In these experiments we focus on comparing joint mappings of the speech signal and articulatory features from one speaker to another and how they compare to mappings that use only speech signal features.

4.2 MOCHA Database

The particular articulatory position data investigated in this document is the freely available MOCHA database [Wrench, 1999], which includes recordings of the 460-sentence

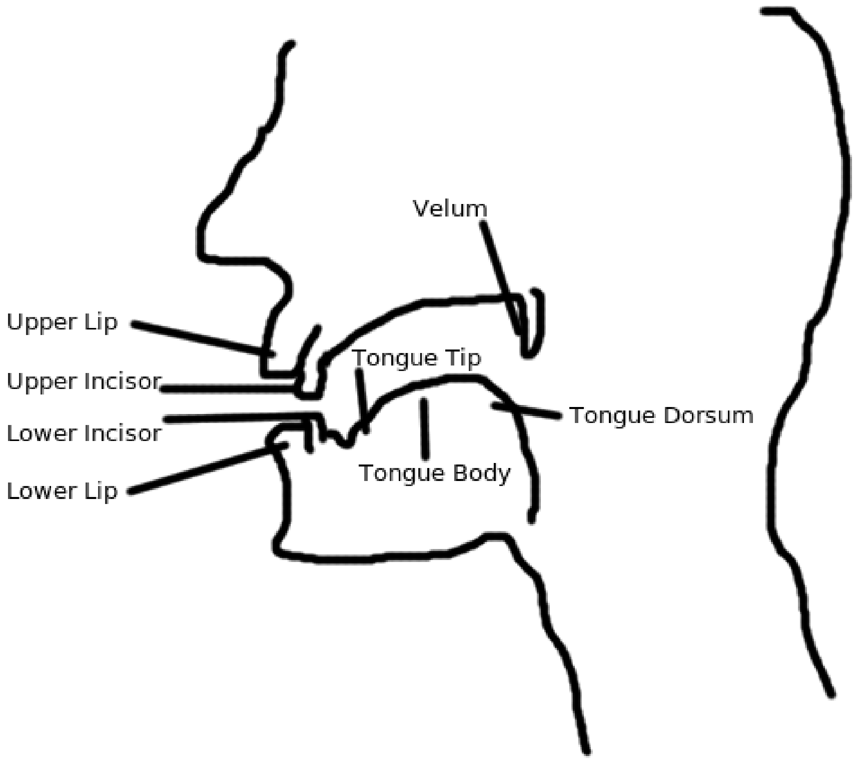


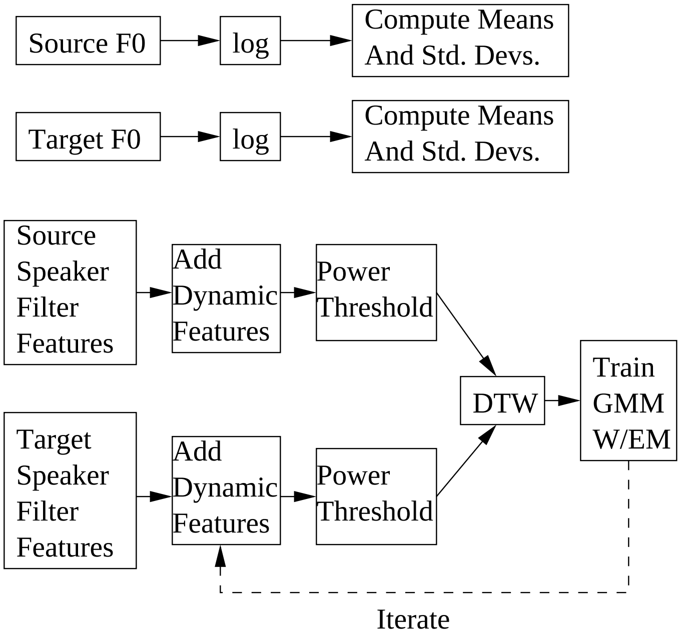
Figure 4.2: Articulatory Positions Recorded in the MOCHA Database

British TIMIT corpus along with coordinates in the midsagittal plane for the upper and lower lip, the lower incisor, three points on the tongue, and the velum of each speaker. These points are depicted in Figure 4.2, which represents the midsagittal view of the articulators of a person facing to the left.

The MOCHA database also supplies laryngograph files and electropalatograph files, which were not used in our experiments. At the time the following experiments were conducted, full data was available for two speakers, labeled msak0 and fsew0. The msak0 speaker is male and has a northern English accent. The fsew0 speaker is female and has a southern English accent.

The following experiments are based on features derived from the audio files and the EMA files. The audio files contain 16 bit samples at a rate of 16kHz. The EMA files contain samples at a rate of 500Hz of the x and y coordinates in the midsagittal plane

Figure 4.3: *Voice Transformation Training*



of the positions of 7 different articulators, for a total of 14 values per sample. These 7 articulators include the upper and lower lip, the lower incisor, three points on the tongue, and the velum. The EMA files also contain additional coordinates for the bridge of the nose and the upper incisor, but they are only used for calibrating the positions of the other articulators and are not used as features in the following experiments.

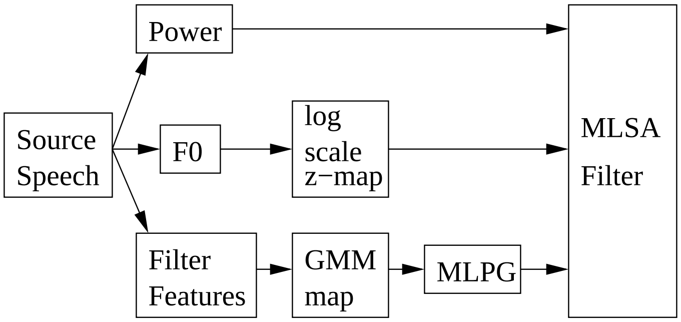
4.3 Baseline Voice Transformation

The baseline voice transformation used in these experiments is based on the FestVox scripts that were described in Chapter 3. For summary and reference, Figure 4.3 depicts the training process, and Figure 4.4 depicts the transformation process.

4.3.1 Error Measure

The same general **Mel-Cepstral Distortion (MCD)** measure that was described in the previous chapter in the context of **Dynamic Time Warping (DTW)** is used in the following experiments as an objective error measure to compare transformed utterances to reference

Figure 4.4: *Voice Transformation*



utterances recorded by the target speaker. MCD correlates with results from subjective listening evaluations and has been used to measure the quality of voice transformation results in other work [Toda et al., 2004a]. MCD is essentially a weighted Euclidean distance, and in this case is calculated on only the MCEP features and not their dynamic features (unlike during the DTW performed during voice transformation):

$$MCD = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(m_d^{(t)} - m_d^{(r)})^2}$$

where $m_d^{(t)}$ is the dth MCEP of a frame of transformed speech, and $m_d^{(r)}$ is the dth MCEP of the corresponding frame in the reference utterance recorded by the target speaker. DTW is used to align the utterances before computing the MCDs, because they will probably differ in length.

MCD is more related to filter characteristics of the vocal tract. Although characteristics such as power and fundamental frequency are also important to the quality of voice transformation output, the use of MCD for these experiments seems appropriate as the articulatory positions are expected to be most closely related to the filter characteristics of the vocal tract.

For the following results, no power thresholding was performed on frames before calculating MCDs, and the transformed MCEPs were used, as opposed to MCEPs rederived after synthesizing waveforms.

4.4 Adding Articulatory Position Data

Numerous experiments were conducted which added articulatory position data to the baseline MCEP features within the same general framework. The baseline voice transformation scripts were modified to allow the use of articulatory position features instead of or in addition to MCEP features. The rest of the processing continued in the same basic manner, with the exception that the error measure for the combination of articulatory position data and MCEPs was based solely on the MCEP subset. In the following descriptions, EMA will be used to refer to the articulatory position data, because it is the abbreviation for Electro-Magnetic Articulograph, which is the specific type of articulatory position data that we used. Similarly, EMAMCEP will be used to refer to the combined use of EMA and MCEP data.

The EMA data from the MOCHA data had to be processed before combination with the MCEPs because it was sampled every 2ms instead of every 5ms, and the durations of the EMA files did not always match the durations of the audio files. Resampling was performed with the ch_track program from the Edinburgh Speech Tools [Taylor et al., 1998], and EMA or MCEP features were truncated when the lengths didn't match.

The experiments include transformations from each of the two MOCHA speakers, msak0 and fsew0, to the other. The data was split into a training set of 414 utterances and a test set of 46 utterances. Most of the voice transformations were trained on a subset of 50 utterances due to the amount of time necessary to process the entire training set and the similarity of the results between using the entire training set and the subset in some preliminary experiments.

Finally, there were some additional pragmatic considerations for the GMM training. The original EMA values were measured in thousandths of centimeters, and in some cases exceeded 5,000. Using these original values led to overflow errors with the training program, so we z-scored the EMA values to put them in a manageable range. Also, the number of Gaussian components in the GMM could affect whether training succeeded. In some cases the training program was unable to estimate parameters for the GMM due to a lack of parameter convergence during the EM training. In the following tables, the results for such trials will be marked as **N/A (Not Applicable)**.

We experimented with multiple numbers of Gaussian components in the mixture models to determine a range of success and also to track where increasing the number of components improved performance. After the initial trials, our basic choices were 16, 32, 64, or 128 components. These generally appeared to capture the range where results first improved and then worsened, presumably due to overtraining, or training even failing.

Table 4.1: *MCEP vs. EMAMCEP MCD Means (Std. Devs.)*

	msak0 to fsew0		fsew0 to msak0	
M	MCEP	EMAMCEP	MCEP	EMAMCEP
1	6.33(1.62)	6.88(1.61)	5.59(1.59)	5.95(1.68)
2	5.84(1.95)	6.34(1.97)	5.51(1.59)	5.79(1.71)
4	5.67(1.94)	6.25(2.06)	5.57(1.42)	5.81(1.64)
8	5.74(1.78)	6.60(1.65)	5.31(1.55)	5.95(1.62)
16	**5.58(1.79)**	6.09(1.89)	5.20(1.58)	5.46(1.62)
32	5.74(1.79)	N/A	5.06(1.62)	5.66(1.50)
64	5.74(1.70)	N/A	**5.01(1.63)**	N/A
128	N/A	N/A	N/A	N/A

4.5 Experiments

4.5.1 Baseline Experiments

The first experiment was a comparison of only using MCEP features with using a combination of MCEP and EMA features. The only change made to the GMM mapping procedure for the initial trials including EMA was to include the EMA values in the feature vectors as well as the MCEP values. Performance was measured by calculating the MCD between transformed utterances and recorded reference utterances from the associated target speaker, based only on the 1st through 24th MCEP features. The means and standard deviations of the Mel-cepstral distortions are in Table 4.1. Smaller values are better. The best result among the msak0 to fsew0 trials and the best result among the fsew0 to msak0 trials are both indicated in boldface. MCD means and standard deviations will also be used to report results in the rest of the tables in this chapter.

Adding all the EMA features directly as z-scored x and y coordinates in the midsagittal plane did not help in any of the trials, so it was necessary to investigate the data and the learning process more closely.

4.5.2 Attempts to Remove Noise from the Data

One possible cause for the lack of improvement was the presence of noise in the EMA data. Some potential causes were:

- The electrical apparatus originally used to collect the data

- The alignment of the MCEP with the EMA
- The resampling of the EMA data to match the default MCEP sampling rate

It has been noted by others [Uraga and Hain, 2006] that there appears to be line noise at 50Hz in the MOCHA data. For that reason and also assuming that the motions of the articulators would be slow enough at our sampling rate, we tried applying two different low-pass filters with cut-offs of 45Hz and 10Hz to the MOCHA data using the sigfilter program from the Edinburgh Speech Tools [Taylor et al., 1998]. For both cut-offs, adding the low-pass filtered EMA data to the MCEP data failed to reduce the average MCD when compared to only using the MCEP data for voice transformation.

Another possible problem with the MOCHA data is that the means of the feature positions appear to vary over time more than what would be expected based on the differing phonetic contexts alone, according to other researchers [Richmond, 2001] [Shiga, 2005]. Although these sources were not certain whether this "drift" came from the Electo-Magnetic Articulograph or the adjustment of speakers to the probes used to measure them, they found for their tasks that it was useful to try to compensate for it. We tried applying the "drift correction" strategy from the latter reference to the EMA data. This consisted of treating the mean values per utterance of the EMA features as signals, low-pass filtering these signals forward and backward with a FIR filter of length 100 and cut-off of 0.04π radians, and subtracting the resulting per-utterance "drift" values from the corresponding EMA features in the corresponding utterances. Adding the resulting drift-corrected data to the MCEP data failed to reduce the MCD error when compared to using the MCEP data alone for voice transformation, as can be seen by comparing the "Drift" columns in Table 4.2 and Table 4.3 with the "Baselines" columns, which repeat the appropriate baseline figures from Table 4.1 for reference. Table 4.2 gives figures for transformations from the fsew0 speaker to the msak0 speaker, and Table 4.3 gives figures for transformations from the msak0 speaker to the fsew0 speaker.

Another possible problem was that the EMA data was not aligned with the MCEP data. We experimented by shifting the EMA data one frame by repeating the first EMA frame. The results of these experiments are in the "Delay" columns of Table 4.2 and Table 4.3.

This only made a minor change to the results and demonstrated that shifting the EMA by repeating the first EMA frame did not help. A companion experiment was performed where the first EMA frame was removed from each utterance. Shifting the EMA frames in that direction did not lead to an improvement in the results for trials using EMA data either. The results of these experiments are in the "Advance" columns in Table 4.2 and Table 4.3. In both of these experiments, due to differences in the truncation of the feature files after alignment, there are small differences in the results for the trials which only used

Table 4.2: *VT (fsew0 to msak0) with EMA: Noise Compensation: MCD Means (Std. Devs.)*

	Baselines		Drift	Delay	Advance
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	EMAMCEP
16	5.20(1.58)	5.46(1.62)	5.58(1.59)	5.49(1.62)	5.47(1.59)
32	5.06(1.62)	5.66(1.50)	5.31(1.78)	5.45(1.71)	5.69(1.67)
64	**5.01(1.63)**	N/A	N/A	N/A	N/A
128	N/A	N/A	N/A	N/A	N/A

Table 4.3: *VT (msak0 to fsew0) with EMA: Noise Compensation: MCD Means (Std. Devs.)*

	Baselines		Drift	Delay	Advance
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	EMAMCEP
16	**5.58(1.79)**	6.09(1.89)	6.09(1.73)	6.16(1.84)	6.15(1.79
32	5.74(1.79)	N/A	N/A	N/A	N/A
64	5.74(1.70)	N/A	N/A	N/A	N/A
128	N/A	N/A	N/A	N/A	N/A

MCEP data.

4.5.3 Attempts to Refine the Transformation Process

The baseline script that was used to perform voice transformation was based on techniques that were refined over time to handle MCEP data. It was unclear whether parts of this process were still appropriate when adding EMA data to the MCEP vectors. We investigated the following areas more closely:

- Dynamic Time Warping (DTW) used for alignment of the two speakers
- Use of the Maximum Likelihood Parameter Generation (MLPG) algorithm
- Use of multiple iterations of DTW during training

In the baseline voice transformation system, DTW was performed over all features and their derived dynamic features to align feature vectors between speakers. Using a distance metric that treated all features equally did not seem appopriate, because the MCEP and

Table 4.4: *VT (fsew0 to msak0) with EMA: Process Changes: MCD Means (Std. Devs.)*

	Baselines		MCEP DTW	MCEP DTW/no MLPG	
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	MCEP
16	5.20(1.58)	5.46(1.62)	5.35(1.73)	4.97(1.86)	4.95(1.57)
32	5.06(1.62)	5.66(1.50)	5.31(1.77)	4.97(1.83)	4.91(1.59)
64	5.01(1.63)	N/A	N/A	N/A	5.10(1.69)
128	N/A	N/A	N/A	N/A	N/A

Table 4.5: *VT (msak0 to fsew0) with EMA: Process Changes: MCD Means (Std. Devs.)*

	Baselines		MCEP DTW	MCEP DTW/no MLPG	
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	MCEP
16	5.58(1.79)	6.09(1.89)	5.84(1.81)	5.49(1.56)	**5.39(1.78)**
32	5.74(1.79)	N/A	5.90(1.76)	**5.50(1.81)**	**5.60(1.78)**
64	5.74(1.70)	N/A	N/A	N/A	5.76(1.84)
128	N/A	N/A	N/A	N/A	N/A

z-scored EMA values were not of the same scale. For this reason, we ran experiments that only considered the MCEP values during DTW when additional EMA features were used. The results are in the "MCEP DTW" columns of Table 4.4 and Table 4.5. Table 4.4 gives figures for transformations from the fsew0 speaker to the msak0 speaker, and Table 4.5 gives figures for transformations from the msak0 speaker to the fsew0 speaker. In both tables, the "Baselines" columns repeat the relevant baseline MCEP and EMAMCEP figures from Table 4.1 for comparison. As can be seen, this approach did not give better results than using MCEP data alone for the entire process. It did, however, improve over the baseline EMAMCEP performance, so this technique was retained in later experiments.

One other thing to note is that basing the DTW only on MCEP features in the trials that also include EMA data leads to the same source speaker and target speaker frames being aligned across the different trials. This is not guaranteed when the DTW in the trials using EMA data also uses EMA values, because the additional EMA values will be included in the calculations of the distances between frames and will change the values in the DTW matrix.

In the baseline voice transformation system, the MLPG program, mentioned in the previous chapter, is used to take the GMM estimates of the target speaker's MCEP and MCEP dynamic feature means and covariances to try to estimate final MCEP values that form a good path. It was unclear whether including EMA features in this process was appropriate. We ran another set of experiments where we used the means of the MCEP features for predictions and did not use MLPG (in addition to using the abovementioned strategy of only considering MCEP and MCEP dynamic feature values during DTW). The results of these experiments are in the "MCEP DTW/no MLPG" columns of Table 4.4 and Table 4.5. Adding EMA data helped in the trial that used 32 Gaussian components for the transformation from msak0 to fsew0, which can be seen by comparing the boldface entries in the table row for 32 Gaussian components. However, this was not a global best result for this transformation direction as the 16 Gaussian trial using only MCEP data still had better results. This global best result is also indicated in boldface.

After applying a number of techniques to refine the voice transformation process, we were able to improve the results including EMA values over the straightforward extension that was considered in the baseline strategy which included EMA values, but the best overall results still came from using MCEP values alone. This led us to try other approaches involving the representation of the EMA features.

4.5.4 Representation of EMA Features

Another possible problem with the previous experiments was that the x and y coordinates in the EMA data may have been a poor match for voice transformation in general or even the GMM mapping technique in particular. Perhaps there is more relevant information in features that are derived from these coordinates. After all, the x and y coordinate values are related to each other, both in terms of pairs being related to the same articulators, and in the sense that the positions of some articulators can pose constraints on the positions of others. Furthermore, the positions of some articulators relative to others provide information on constrictions in the vocal tract, which influence the filter characteristics. We investigated the following types of derived EMA features:

- Distances between the lips
- 1st order differences
- Projections onto lines of best-fit for each articulator

One type of vocal tract constriction that seemed reasonable to measure from the 7 articulators available in the MOCHA database was the distance between the lips. The two-dimensional Euclidean distance between the lips was used as a derived feature. The results for this experiment are in the "Lip Distance" columns of Table 4.6 and Table 4.7, with the "MCEP DTW/no MLPG" results from Table 4.4 and Table 4.5 serving as new,

Table 4.6: *VT (fsew0 to msak0) with EMA: Derived Features: MCD Means (Std. Devs.)*

	MCEP DTW/no MLPG		Lip Distance	2-D Distances	Projection
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	EMAMCEP
16	4.95(1.57)	4.97(1.86)	5.40(1.78)	5.21(1.73)	5.01(1.85)
32	4.91(1.59)	4.97(1.83)	5.25(1.80)	5.14(1.80)	5.00(1.86)
64	5.10(1.69)	N/A	5.19(1.81)	N/A	N/A
128	N/A	N/A	5.19(1.89)	N/A	N/A

Table 4.7: *VT (msak0 to fsew0) with EMA: Derived Features: MCD Means (Std. Devs.)*

	MCEP DTW/no MLPG		Lip Distance	2-D Distances	Projection
M	MCEP	EMAMCEP	EMAMCEP	EMAMCEP	EMAMCEP
16	**5.39(1.78)**	5.49(1.56)	5.64(1.96)	5.47(1.99)	5.60(1.78)
32	5.60(1.78)	5.50(1.81)	5.55(2.00)	5.62(2.01)	**5.36 (1.97)**
64	5.76(1.84)	N/A	6.07(2.08)	**5.56(2.02)**	N/A
128	N/A	N/A	6.01(2.11)	N/A	N/A

improved baselines for comparison. Although adding the single feature of lip distance to the MCEPs gave a better MCD than the original baseline when transforming from the msak0 voice to the fsew0 voice with 32 Gaussian components in the GMM, it did not improve upon the new MCEP baseline.

Another hypothesis was that capturing information about the motion of the articulators in two-dimensional space might supply more information. We ran experiments where the two-dimensional Euclidean distances were calculated between (x, y) coordinate pairs from frame to frame. This constructed 7 EMA derived features that could be added to the MCEP data. These trials were performed using only the MCEP and MCEP dynamic features for DTW and did not use MLPG. The results of these experiments are in the "2-D Distances" columns of Table 4.6 and Table 4.7. As can be seen by comparison with the new MCEP baseline, adding these EMA derived distance features helped in the case of using 64 Gaussian components for the transformation from msak0 to fsew0. This result is indicated in boldface in the table. However, this was not a global positive result for the msak0 to fsew0 transformation as it did not perform as well as the 16 and 32 Gaussian component trials which only used MCEP data.

One problem with using 2-dimensional distances as features is that it does not include any notion of directionality, which seems like it should be important. There is a question of how to include this directionality in a meaningful way in the vectors used in the

GMM mapping strategy. Although the articulator positions were measured in two dimensions, in many cases it appeared that individual articulators moved more along certain directions than others. For example, the lower incisor data showed more motion along the y-dimension than the x-dimension. In an attempt to capture some of this information, we derived features from the EMA data by running linear regression on the (x, y) coordinate pairs in the training set for individual articulators to create best-fit lines, projecting the EMA (x, y) pairs onto these lines, and determining how far along these lines the articulators were. This is similar to using the first component from **Principal Component Analysis (PCA)** [Pearson, 1901]. The results of using these projected EMA features are in the "Projection" columns of Table 4.6 and Table 4.7. Again, in these trials, only the MCEP features were used for DTW and MLPG was not used. By comparison with the new MCEP baseline figures in the MCEP columns, it can be seen that not only does adding the projected EMA features improve the trial using 32 Gaussians for the transformation from msak0 to fsew0, but that this is a global positive result as it is better than all the other trials for transforming msak0 to fsew0, including the ones that only use MCEP data. Finally, using a strategy that combined a modification of the voice transformation, in addition to a change in the EMA feature representation led to a positive result, in the sense that the average MCD was smaller. This result, however, was not statistically significant. We collected the average MCDs over the frames for each test utterance produced by these two systems, and performed a paired t-test. The null hypothesis, that the data came from the same distribution, could not be rejected at a 5% level of significance.

A different approach to investigating the possibility of the data being a mismatch for the model is to switch the model instead of changing the features. To this end, we tried using wagon, the Classification And Regression Tree (CART) program from the Edinburgh Speech Tools [Taylor et al., 1998], instead of GMM mapping and MLPG smoothing, to perform the mapping between speakers. Using a step size of 100, CART predicted MCEPs from MCEPs in the fsew0 to msak0 direction with a MCD mean of 4.71 and standard deviation of 1.71. Using the combination of EMA data with MCEPs from the fsew0 speaker to predict MCEPs for the msak0 speaker gave a MCD mean of 5.22 and standard deviation of 1.90. Even with a different learning algorithm, adding EMA data failed to help improve voice transformation in terms of MCD. Although the numbers for the individual trials were better than for the GMM mapping baseline, there was the same general trend of the MCEP-only trial performing better than a trial that added EMA x and y coordinates directly.

4.6 Summary and Conclusions

A number of strategies were applied to the problem of trying to use EMA data to improve a fairly standard GMM mapping based voice transformation technique in terms of Mel-Cepstral Distortion. For the most straightforward extension of the baseline voice transformation technique, none of the experimental trials that used additional EMA data directly as x and y coordinates improved the Mel-Cepstral Distortion. We made a number of attempts to use the EMA data to improve results. These attempts focused on the following three areas:

1. Removing noise from the data
2. Modifying parts of the voice transformation process that no longer appeared appropriate when using a combination of EMA and MCEP data
3. Finding a better way of representing EMA information in the model

In the first case, attempts to remove noise through filtering and realigning the EMA data, among other things, did not appear to help. In the second case, changing the way DTW was performed and not using MLPG led to results for the trials that used EMA to improve to the point where there was a trial where adding the EMA data led to better performance than using MCEP data alone. However, this was still not a global positive result as there was an MCEP trial with a different number of Gaussian components that outperformed it. In the third case, there was another positive result that came from using the distance between the lips, and finally, the first global positive result appeared in the case of using features derived from EMA by projecting the coordinates onto lines fit to the data through linear regression. In this case, the strategies of basing the DTW only on the MCEP data and not using MLPG were also followed. As this single global positive result was not statistically significant, this approach was not considered successful.

It appears that the use of EMA data to improve voice transformation is not very straightforward. One additional thing to note is that all of the positive results occurred while transforming from msak0 to fsew0. There were none in the other direction. This appears to be another case of asymmetry in voice transformation. Asymmetric results have also been noted in identity perception for voice transformation [Toth and Black, 2006].

There are numerous areas for further investigation. Maybe the Mel-Cepstral Distortion metric is not good enough for this task, even though it shows some correlation to subjective listening tests. Perhaps the information necessary for voice transformation is already present in MCEPs and EMA provides nothing additional. It is also possible that EMA features need to be combined or represented in a different space before they will be useful. Further experimentation will be necessary to tell.

Chapter 5

Cross-Speaker Articulatory Positions

5.1 Introduction

One of the problems with using articulatory position data, such as that found in the MOCHA database, is that it is difficult to collect. In addition to requiring access to an Electro-Magnetic Articulograph, it requires attaching probes to a person's articulators and, for older models of the device, restraining the person's head movements. As a result, most of the experiments performed with it have been very small scale experiments limited to a few speakers. If articulatory position data is to become useful for speech applications in general, it either needs to become much easier to collect, or some way of leveraging existing articulatory position data for use with other speakers must be found. In this chapter, we explore the latter technique to create what we call **"cross-speaker articulatory features"** or **"pseudo-articulatory features"**, which, in the present formulation, are EMA feature predictions based on MCEP features. These features can be seen as a different representation of Mel-cepstral coefficients.

The first part of this chapter verifies the plausibility of cross-speaker pseudo-articulatory features by demonstrating that they can be helpful in the external task of phonetic feature prediction.[1] The second part then applies these features to voice transformation.

[1] The sections in this chapter on phonetic feature prediction are adapted from a conference paper [Toth, 2005].

5.2 Acoustic, Articulatory, and Phonetic Features

The primary parametrizations of speech used in automatic speech recognition and synthesis are based on DSP techniques. MFCC, LPCC, and derived features can be readily extracted from acoustic signals and allow the construction of relatively high-performance speech systems. However, these features (though related) are a bit removed from the actual physical process of speaking. When a person speaks, the produced sound is the result of respiration and voicing, combined with the motions of articulators, which affect the shape of the vocal tract. The locations of these articulators should also be useful for the parametrization of speech, and should enable the construction of new models. So far, EMA data has been used to perform a variety of experiments. Some concern relationships between articulatory positions and acoustic features derived from speech signals [Richmond, 2001] [Hiroya and Honda, 2002] [Shiga and King, 2004] [Toda et al., 2004a] [Toda et al., 2004b] Toda et al. [2008]. Others use articulatory positions to aid in speech recognition [Markov et al., 2003] [Markov et al., 2004].

At the same time, there have been other lines of work concerned with what have traditionally been called "acoustic-phonetic" features [Rabiner and Juang, 1993], but are occasionally referred to as "articulatory" features [Metze and Waibel, 2002] [Frankel et al., 2004] [Wester et al., 2004]. These features are categorial and describe phones when taken together. Some examples include voicing and placement of articulation. To minimize confusion, we will refer to such features as "phonetic" features in this document. Recent work has included an attempt to go beyond the "beads-on-a-string" approach to modeling speech [Ostendorf, 1999] to models based on parallel streams of phonetic features. Such an approach has been demonstrated to improve speech recognition [Metze and Waibel, 2002].

As many of the traditional phonetic features are related to notions of placement in the vocal tract, it seems natural to consider the connection between them and actual positions of articulators as measured by an EMA. The following sections discuss a number of experiments investigating this relationship. It is hoped that mappings from articulatory positions to phonetic features will enable the extension of current speech models and the construction of new ones.

5.3 Predicting Phonetic Features from Articulatory Positions

5.3.1 Phonetic Features

In order to predict phonetic features from articulatory positions, it is first necessary to determine which phonetic features to predict. One strategy is to use a set of multi-valued features, such as the manner, place, voicing, rounding, front-back, and static features described in a paper by Frankel et al. [2004]. A potential complication of this approach is that such multi-valued phonetic features are typically conceived of in a hierarchical manner. For instance, some features such as high and low are typically considered only for vowels, while other features such as labial and velar are typically considered only for consonants. In the paper by Frankel et al. [2004], all of these values are possible for the place feature. The model used in the paper approaches this problem by conditioning the place value on the manner value, which can be vowel, silence, or one of a number of consonant types. Without some sort of hierarchy, though, place values associated with vowels may be confusable with place values for consonants. This may degrade performance.

Another strategy is to use a set of binary features that are either present or absent as in the speech recognition work by Metze and Waibel [2002]. In this approach, a hierarchy of features is not necessary, but one potential complication is that many more features are needed to describe the phone set, and the cross-product of the values can be quite large. Based on how the features are used, however, this may not be a problem.

5.3.2 Articulatory Position Features

After selecting the phonetic features, it is necessary to decide which articulatory position features to use. Again, we used data from the msak0 and fsew0 speakers from the MOCHA database [Wrench, 2001].

5.3.3 Model

Stepwise CART [Hocking, 1976] was used to construct models for predicting the 18 binary phonetic features listed in the first column of Table 5.1 from the articulatory positions. This was fewer than the full set of 76 binary features used in the paper by Metze and Waibel [2002] but sufficient for the purpose of demonstrating a relationship between the phonetic and articulatory position features.

Stepwise CART was chosen as a model because it can ignore predictor features when it does not find a high correlation with the predictee. This was considered important because it is believed that the positions of some articulators may be irrelevant to the values of some phonetic features. For example, the position of the velum is probably unrelated to the labial binary phonetic feature. The stop-size for the trees was determined by cross-validation. For each speaker, 8/10 of the utterances were used for training, with an additional 1/10 used as a held-out set for the stepwise processing. The remaining 1/10 were used for testing. A few utterances were not used due to corrupt data. During training, as suggested by Metze and Waibel [2002], only the center frames of the phones were used in order to minimize the effects of co-articulation. The centers of the phones were derived by automatically labeling the boundaries with SphinxTrain [Carnegie Mellon University, 2001]. The center of each phone was labeled with the phone's canonical phonetic features.

Other work [Metze and Waibel, 2002] has used MFCCs to predict binary phonetic features. This work used different corpora that weren't "phonetically balanced" like the MOCHA data and only provided overall accuracies for the phonetic feature recognizers, so the results cannot be compared. However, this work does demonstrate that MFCCs have predictive value for phonetic features.

As our baseline voice transformation system uses MCEPs, which are approximations to MFCCs, we decided to conduct experiments to predict phonetic features from MCEPs and from a combination of articulatory positions and MCEPs. Because MCEPs are readily derived from the speech signal, using articulatory positions to predict phonetic features would only be useful in cases where the performance was improved or the speech signal was not available. The trials in the following experiments used the 0th through 24th MCEPs.

5.3.4 msak0 Phonetic Feature Prediction Results

The results of the trials for the msak0 utterances from the MOCHA database are listed in Table 5.1. The listed results are F-scores that were derived by combining precision and recall according to the formula:

$$F = \frac{2 * p * r}{p + r} \tag{5.1}$$

where F is the F-score, p is the precision, and r is the recall. χ^2 tests were used to compare the true-positive/false-positive/true-negative/false-negative breakdowns for each phonetic feature between the MCEP trials and the other trials. Boldface entries in the "EMA" and "Both" columns represent trials where the breakdowns were different from

Table 5.1: *msak0 Binary Phonetic Feature Prediction F-scores*

Feature	MCEP	EMA	Both
unvoiced	0.683	**0.203**	0.683
stop	0.386	**0.254**	**0.573**
vowel	0.511	**0.407**	0.511
lateral	0.028	*0.136*	*0.136*
nasal	0.280	0.234	0.287
fricative	0.447	0.507	0.515
labial	0.175	**0.457**	**0.457**
palatal	0.037	*0.368*	*0.037*
velar	0.088	**0.550**	**0.408**
glottal	undef.	*undef.*	*undef.*
high vow.	0.270	**0.132**	**0.132**
mid vow.	0.205	0.197	0.205
low vow.	0.333	0.201	0.259
front vow.	0.198	0.184	**0.406**
back vow.	0.062	**0.141**	0.062
diphthong	0.072	0.182	0.072
round	0.154	0.139	0.256
alv. fric.	0.586	**0.338**	0.601

the corresponding trials in the “MCEP” column at a significance of 0.05. Italic entries in the “EMA” and “Both” columns represent trials where the χ^2 test was not considered valid because at least one of the breakdowns for either the trial in that column or the corresponding trial in the “MCEP” column had fewer than 5 examples.

For the 18 features that were tried, 5 were better predicted from MCEPs (unvoiced, vowel, high vowel, mid vowel, low vowel), 6 were better predicted from articulatory positions (lateral, labial, palatal, velar, back vowel, diphthong), and 6 were better predicted from a combination of the two (stop, nasal, fricative, front vowel, round, alveolar fricative). While predicting the glottal feature, none of the approaches had a **true-positive**, which is a prediction that the feature value is true when it actually is true, so the prediction of the glottal feature was considered unsuccessful.

The experimental results demonstrate that the prediction of some phonetic features was indeed improved by using articulatory positions as predictors. Most of the features that were better predicted by articulatory positions were related to placement, which was expected. The MCEPs were much better at predicting whether a phone was unvoiced. This is not surprising because voicing is controlled by the larynx, which was not treated as an articulator in these experiments.

5.4 Cross-Speaker Articulatory Positions

As mentioned previously, articulatory position data would be more useful if there were a way to use it with speakers for whom it has not been collected. To these ends, we experimented with approaches to map from one speaker's MCEPs to another speaker's articulatory positions and back. Then these predicted articulatory positions could be used in other models.

5.4.1 Corpora

In order to map between two speakers, we needed data from them. For our initial cross-speaker experiments, we chose the previously mentioned msak0 data and the *Facts and Fables* (FAF) data [Zhang et al., 2004]. The FAF data is quite different from the msak0 data. The FAF corpus consists of 107 utterances of paragraph or multi-paragraph length which contain a total of around 14,000 words. The utterances consist of public domain text from Project Gutenberg [Hart]: excerpts from *Aesop's Fables* and the *CIA World Factbook (2000)*. The speaker was a male with a Midwestern American accent. For each utterance, there was a 16-bit acoustic file sampled at 16kHz, but no EMA file. This corpus was created to study prominence and super-sentential prosody, so it is a bit different from the MOCHA msak0 and fsew0 corpora.

5.4.2 Cross-Speaker Mapping Approaches

We experimented with a few novel approaches to map from one speaker's acoustic data to another speaker's articulatory position data. We called these approaches the baseline approach, the z-score mapping approach, and the DTW direct approach.

Baseline Cross-Speaker Mapping

In the baseline approach, the MCEPs of one speaker were treated as being in the same space as those of another, and thus mappings between MCEPs and articulatory positions trained on only one speaker could then be applied to the MCEPs of another.

Z-Score Mapping Cross-Speaker Mapping

In the z-score mapping approach, the MCEPs of one speaker were z-score mapped to the range of the other speaker before single-speaker MCEP-to-articulatory-position mappings were applied.

DTW Direct Cross-Speaker Mapping

In the DTW direct approach, DTW was first used to align the source and target speaker utterances using Euclidean distances between the MCEPs with the Itakura rule Itakura [1975]. Aligning the utterances was necessary, because they typically had different lengths. After the alignment, mappings were learned directly between the MCEPs from the source speaker's frames and the articulatory positions from the target speaker's aligned frames.

5.4.3 Cross-Speaker MCEP/Articulatory Position Results

Table 5.2: *Cross-Speaker MCEP/Articulatory Position Mappings*

	Baseline	Z-Score	DTW
FAF MCEP to msak0 EMA	RMSE (mm)		
Lin. Reg.	2.30	2.13	2.26
CART	2.49	2.21	2.23
msak0 EMA to FAF MCEP	MCD mean ± std		
Lin. Reg.	9.43 ± 2.73	7.63 ± 2.29	7.90 ± 3.05
CART	9.48 ± 2.78	7.87 ± 2.40	7.89 ± 3.16
FAF MCEP to msak0 EMA to FAF MCEP	MCD mean ± std		
Lin. Reg.	9.38 ± 2.44	7.27 ± 2.13	7.40 ± 2.55
CART	10.03 ± 2.46	9.92 ± 2.43	7.41 ± 2.69

The three cross-speaker mapping approaches were tried using both linear regression and CART for the mappings between MCEPs and articulatory positions. The mappings were performed between utterances from the *Facts and Fables* (FAF) database and the msak0 speaker from the MOCHA database. Because the *Facts and Fables* text was different, a unit-selection synthesizer based on the *Facts and Fables* recordings was used to

Table 5.3: *Cross-Speaker Mappings vs. Single-Speaker Mappings*

	Cross-Speaker (FAF and msak0)	Single-Speaker (msak0)
MCEP to EMA	RMSE (mm)	
Lin. Reg.	2.13	2.07
CART	2.21	1.95
EMA to MCEP	MCD mean ± std	
Lin. Reg.	7.63 ± 2.29	6.14 ± 2.63
CART	7.87 ± 2.40	5.61 ± 2.49
MCEP to EMA to MCEP	MCD mean ± std	
Lin. Reg.	7.27 ± 2.13	5.54 ± 2.44
CART	9.92 ± 2.43	5.30 ± 2.23

produce British TIMIT utterances to match the msak0 data. The results are reported in Table 5.2. Average RMSE per articulator is used as the error metric for trials that predict articulatory positions, and Mel-Cepstral Distortion (MCD) mean and standard deviation are used as the error metric for MCEPs. These measures are used and described in [Toda et al., 2004a] [Toda et al., 2004b].

Table 5.3 compares the best cross-speaker results from Table 5.2 with the results from using only msak0 data, but there are some inherent difficulties. For the mappings from FAF MCEPs to msak0 articulatory positions, it is possible to compare the results to single-speaker mappings from msak0 MCEPs to msak0 articulatory positions, but it is harder to determine what the true values should be. Questions arise such as: "Where should one person's articulators be when another person speaks?" For the mappings from msak0 articulatory positions to FAF MCEPs, there are similar considerations. When considering the "roundtrip" mapping from FAF MCEPs to msak0 articulatory positions and back to FAF MCEPs, there is a notion of truth for the final result, because we want the output of the composed map to match the input, but that alone is not sufficient for good results, because we would like the intermediate results to behave like articulatory positions. It would be possible to construct an identity map that would give perfect end results, but not produce anything useful for articulatory positions. For these reasons, it would be good to have another measure of the quality of articulatory position predictions. If there is another quantity that is correlated with articulatory positions, this may potentially be used as a measure.

5.5 Cross-Speaker Phonetic Feature Prediction

Phonetic feature prediction is one possible candidate for measuring the usefulness of cross-speaker articulatory position prediction because articulatory positions have been demonstrated to be useful for predicting some phonetic features in the single-speaker case.

We investigated this possibility by conducting experiments using the fsew0 and FAF data to predict msak0 articulatory positions, which were then used to predict phonetic features.

5.5.1 fsew0 Phonetic Feature Prediction

Table 5.4: *fsew0 Binary Phonetic Feature Prediction F-scores*

Feature	MCEP	fsew0 EMA	MCEP+EMA	pEMA	MCEP+pEMA
unvoiced	0.645	**0.356**	0.598	**0.318**	0.681
stop	0.580	**0.198**	0.569	**0.183**	0.550
vowel	0.603	**0.519**	0.653	**0.428**	0.603
lateral	0.060	*0.067*	*0.060*	*undef.*	*0.040*
nasal	0.088	**0.209**	**0.481**	**0.099**	**0.400**
fricative	0.562	0.466	0.539	**0.217**	0.496
labial	0.052	**0.436**	**0.429**	0.097	**0.053**
palatal	0.429	**0.145**	**0.595**	*0.047*	*0.086*
velar	0.136	**0.328**	**0.460**	*0.016*	*0.042*
glottal	0.067	*undef.*	*undef.*	*undef.*	*undef.*
high vow.	0.383	**0.254**	0.339	**0.102**	0.383
mid vow.	0.273	0.194	0.273	0.197	0.273
low vow.	0.298	0.377	0.298	**0.262**	0.411
front vow.	0.379	0.446	0.451	**0.130**	0.310
back vow.	0.206	**0.082**	0.206	0.130	0.206
diphthong	0.047	*0.163*	*0.047*	*0.081*	*0.045*
round	0.086	*0.052*	0.086	0.058	*0.027*
alv. fric.	0.705	**0.514**	0.680	**0.269**	0.705

For the first cross-speaker phonetic feature prediction experiments, pseudo-articulatory positions were created and used to predict fsew0 phonetic features. The pseudo-articulatory positions were created by using the z-score mapping technique to predict msak0 articulatory positions from fsew0 MCEPs. These pseudo-articulatory positions and their associated fsew0 phonetic features were split into a training set, consisting of 90% of the data, and a test set, containing the other 10%. The training set was used to learn decision trees which predicted phonetic features from the pseudo-articulatory positions. The pseudo-articulatory features from the test set were then used with these decision trees to predict phonetic features. The results are compared to prediction of fsew0 phonetic features based on actual fsew0 articulatory positions in Table 5.4. The results listed in the EMA column were for predictions from actual fsew0 articulatory positions from the 7 EMA (x,y)-coordinate pairs. The results listed in the pEMA column were predicted from articulatory position predictions for the msak0 speaker based on the fsew0 MCEPs using the z-score mapping cross-speaker approach. Again, the reported results are F-scores based on precision and recall, and χ^2 tests were used to compare the true-positive/false-positive/true-negative/false-negative breakdowns for each phonetic feature between the MCEP trials and the other trials. Boldface entries in the columns to the right of the "MCEP" column represent trials where the breakdowns were different from the corresponding trials in the "MCEP" column at a significance of 0.05. Italic entries in the columns to the right of the "MCEP" column represent trials where the χ^2 test was not considered valid because at least one of the breakdowns for either the trial in that column or the corresponding trial in the "MCEP" column had fewer than 5 examples.

For the cases using actual fsew0 articulatory positions, four phonetic features were best predicted by articulatory position data alone (lateral, labial, low vowel, diphthong). This was similar to the msak0 trials in Table 5.1 where lateral, labial, and diphthong were also best predicted by articulatory position data alone. Although palatal and velar were best predicted by articulatory positions alone for msak0, they were best predicted by the combination of articulatory positions and MCEPs for fsew0. Of the phonetic features best predicted by articulatory position alone for msak0, only back vowel was best predicted by MCEPs alone for fsew0. However, actual articulatory data predicted low vowel better than MCEPs for fsew0, which was not the case for msak0.

Considering the cases that used cross-speaker articulatory position predictions, labial, diphthong and round were the only cases where only using cross-speaker predicted articulatory positions was not improved by adding actual fsew0 MCEPs. The combination of cross-speaker predicted articulatory positions and actual MCEPs gave the best results overall for unvoiced and low vowel. In the cases of high vowel, mid vowel, back vowel, and alveolar fricative, this combination tied the best performance, but that was because the MCEPs were responsible for that performance, and the cross-speaker articulatory positions were allowed to be ignored in the CART framework.

5.5.2 FAF Phonetic Feature Prediction

Table 5.5: *FAF Binary Phonetic Feature Prediction F-scores*

Feature	MCEP	pEMA	Both
unvoiced	0.291	**0.237**	**0.237**
stop	0.179	**0.180**	**0.180**
vowel	0.431	0.421	0.431
lateral	0.051	*0.022*	*0.022*
nasal	0.124	0.082	0.082
fricative	0.186	**0.116**	**0.116**
labial	0.083	0.125	0.125
palatal	0.109	0.125	0.125
velar	0.113	*0.051*	0.113
glottal	0.133	*0.111*	*0.111*
high vow.	0.110	0.130	0.130
mid vow.	0.240	0.247	0.247
low vow.	0.168	*0.044*	*0.044*
front vow.	0.124	0.112	0.112
back vow.	0.161	**0.099**	**0.099**
diphthong	0.123	**0.079**	**0.079**
round	0.135	*0.044*	*0.044*
alv. fric.	0.096	*0.045*	*0.045*

For the next round of cross-speaker phonetic feature experiments, pseudo-articulatory positions were created and used to predict FAF phonetic features. These pseudo-articulatory positions were created by using the z-score mapping technique to predict msak0 articulatory positions from the FAF MCEPs. Again, the pseudo-articulatory positions were then used to learn decision trees that predicted phonetic features. The F-score results are listed in Table 5.5. Again, χ^2 tests were used to compare the true-positive/false-positive/true-negative/false-negative breakdowns for each phonetic feature between the MCEP trials and the other trials. Boldface entries in the columns to the right of the "MCEP" column represent trials where the breakdowns were different from the corresponding trials in the "MCEP" column at a significance of 0.05. Italic entries in the columns to the right of the "MCEP" column represent trials where the χ^2 test was not considered valid because at least one of the breakdowns for either the trial in that column or the corresponding trial in the "MCEP" column had fewer than 5 examples.

These experiments differed from the fsew0 experiments in that no actual articulatory position data was available for the FAF utterances. Thus the results listed in the “pEMA” and “Both” columns used cross-speaker articulatory position predictions. The cross-speaker articulatory features were better at predicting stop, labial, palatal, high vowel, and mid vowel, and the MCEPs were better at predicting the remaining features. For FAF, there weren’t any cases where the combination outperformed the individual feature sets.

5.6 Discussion

Overall, it appears that articulatory position data can be used to improve the prediction of phonetic features. For one speaker (msak0), the addition of articulatory position data improved the recognition of 12 out of 18 phonetic features. For another speaker (fsew0), its addition improved the recognition of 9 out of 18 phonetic features. There is a considerable degree of overlap between the phonetic features that were best predicted for both speakers by adding articulatory position data.

These experiments introduce some novel techniques for leveraging articulatory position data for use with speakers for whom it has not been collected. One of these approaches was used to predict phonetic features for two speakers (fsew0 and FAF) based on the articulatory position of a third speaker (msak0) and mappings between the speakers’ data. For one speaker (fsew0), adding cross-speaker articulatory positions gave the best results for 2 out of 18 phonetic features. For another speaker (FAF), adding cross-speaker articulatory features gave the best results for 5 out of 18 phonetic features. This demonstrated that cross-speaker articulatory position data can indeed be used to improve phonetic feature prediction.

5.7 Voice Transformation with Cross-Speaker Articulatory Position Features

The ability of cross-speaker articulatory position features to sometimes help in the prediction of phonetic features raises the question of whether they can be used to improve voice transformation. To test this, we constructed pseudo-articulatory features for a pair of female speakers and a pair of male speakers and used them to create pseudo-lip distance features. Then voice transformation trials were run between the speakers in each pair, both using the baseline MCEP features and using the MCEP features with the additional lip distance features. The lip distance feature was chosen because in the experiments in the previous chapter involving real articulatory position data, one of the trials with the lip distance was the first one to demonstrate a positive result for EMA data improving voice transformation.

5.7.1 Data

The speech used in these experiments was from the CMU ARCTIC database [Kominek and Black, 2003]. The female speakers had ids clb and slt, and the male speakers had ids jmk and rms. Speaker jmk had a Canadian English accent, while the other speakers had standard American English accents. For each speaker, 50 utterances from arctic_a0001.wav through arctic_a0050.wav were used for the training set, and 10 utterances from arctic_a0101.wav through arctic_a0110.wav were used for the test set.

MCEP and EMA data was also collected from the two MOCHA speakers, fsew0 and msak0. The training and test sets for the MOCHA speakers were split the same way as before.

5.7.2 Features

Cross-speaker articulatory features were constructed by using the previously described z-score mapping technique with linear regression, as that appeared to be the best of the mappings according to the results of the "round-trip" experiments in Table 5.2. In this way, the MCEPs of the female ARCTIC speakers were mapped to the EMA values of the female MOCHA speaker (fsew0), and the MCEPs of the male ARCTIC speakers were mapped to the EMA values of the male MOCHA speaker (msak0).

After pseudo-articulatory position features were generated for the ARCTIC speakers, distances were calculated between the upper and lower lips, using their predicted x and y coordinates. These pseudo-lip distances were used as features.

5.7.3 Results

The results of the female speaker trials are in Table 5.6, and the results of the male speaker trials are in Table 5.7. These tables list the means and standard deviations of the Mel-cepstral distortions between the transformed utterances and reference utterances from the corresponding target speakers. The M columns in these two tables list the number of Gaussian components in the mixture models, and the MCEP columns list results for trials using

Table 5.6: *Pseudo Lip Distance MCD Means (Std. Devs.)*

	clb to slt		slt to clb	
M	MCEP	pEMAMCEP	MCEP	pEMAMCEP
16	5.38(1.66)	**5.34(1.66)**	5.35(1.87)	5.36(1.85)
32	**5.34(1.72)**	5.39(1.73)	5.32(1.88)	**5.30(1.86)**
64	5.36(1.76)	5.37(1.75)	5.32(1.90)	5.33(1.85)
128		5.39(1.80)		5.32(1.88)

Table 5.7: *Pseudo Lip Distance MCD Means (Std. Devs.)*

	jmk to rms		rms to jmk	
M	MCEP	pEMAMCEP	MCEP	pEMAMCEP
16	6.34(2.77)	6.31(2.79)	6.25(2.69)	6.22(2.67)
32	6.30(2.81)	**6.28(2.80)**	6.20(2.69)	**6.19(2.72)**
64	6.33(2.86)	6.32(2.82)	6.20(2.75)	6.24(2.75)
128	6.41(2.90)	6.36(2.84)	6.32(2.76)	6.31(2.75)

only MCEP features for the spectral mapping. The pEMAMCEP columns list results for trials using both MCEP features and the pseudo-EMA lip-distance feature for the spectral mapping.

For both speaker pairs, and in both transformation directions, the best results came from using lip distances generated from pseudo-articulatory data in addition to the MCEP. Using pseudo-articulatory information did appear to help according to the objective measure, but the improvements were very small and were not statistically significant according to paired t-tests that used MCD averages for each utterance. The null hypotheses could not be rejected at a 5% level of significance.

5.8 Conclusions

There are numerous future directions for this work. One possibility is to see how well articulatory features can predict multi-valued phonetic features. As mentioned earlier, the model would probably need to be augmented to allow for some notion of hierarchy. Another possible direction is to expand the number of articulatory features. Perhaps using positions alone is not enough. It may be more important in some cases to consider distances between different articulators or even features that consider the locations of multiple articulators. Yet another direction is the improvement of cross-speaker mappings. Perhaps the GMM mapping technique from voice transformation could be used to improve cross-speaker mappings by improving the step that takes one speaker's MCEPs to the other's. Finally, there is the question of what can be done with phonetic features. As mentioned in Section 5.1, phonetic features can be used to improve speech recognition performance [Metze and Waibel, 2002]. This and other applications for phonetic feature recognition not only serve as potential benchmarks for the quality of cross-speaker articulatory position predictions, but may demonstrate examples where articulatory position data can be leveraged for general use with speakers for whom it has not been collected.

Chapter 6

Evaluation of Voice Transformation

6.1 Introduction

The previous chapters discussed our efforts to improve voice transformation using MCD as an objective metric. Although lower MCD scores do correlate with better voice transformation quality, neither MCD nor any other automatic voice transformation metric perfectly corresponds to subjective measures, and there is still room for improvement. This chapter investigates some new approaches to measuring voice transformation, both in terms of subjective and objective measures.

6.2 Measuring Voice Transformation

One natural question to ask about voice transformation techniques is how to measure their quality. Intelligibility, naturalness, and speaker recognizability are factors that are commonly measured in the assessment of voice transformation quality [Kain, 2001]. Furthermore, attempts to measure these factors consist both of "objective" and "subjective" tests [Kain, 2001]. Objective tests provide metrics that can be calculated from the output speech and reference speech directly. Subjective tests involve collecting opinions from people in listening experiments and analyzing the results. The strength of objective tests is that they can be performed quickly and automatically. However, when it comes to measuring the quality of voice transformation, the "gold standard" is human perception, and subjective tests are based on it. When objective tests are employed, they are typically used in conjunction with subjective tests and some attempt to correlate the results of the tests is used to justify the objective tests.

6.3 Subjective Measures

Although subjective listening tests have the great advantage of being based on human perception, they are, at their base, subjective.[1] Their results are open to interpretation, and factors which may influence the listeners' opinions must also be taken into account. This paper investigates one such factor: whether knowing the speaker pairs used in voice transformation affects the listeners' opinions in a subjective listening test concerning the speaker recognizability in voice transformation. We proposed a new type of diagram, called a **Transformation Triangle Diagram (TTD)** to aid in visualizing the results of such a subjective listening test.

6.4 Listening Experiment Design

Two groups of people, called Group A and Group B, consisting of speakers and listeners, were selected for a voice transformation listening experiment based on the following criteria:

- Each group had 1 pair of male speakers and 1 pair of female speakers.
- When selecting speakers, priority was given to speakers with similar voices based on our subjective opinions.
- The listeners in each group knew the speakers in their group and did not know the speakers in the other group.

For Group A, the female speakers were **clb** and **slt**, and the male speakers were **ehn** and **ref**. For Group B, the female speakers were **hb** and **jm**, and the male speakers were **mo** and **rf**. Each speaker was recorded reading the first 30 sentences of the CMU ARCTIC corpus [Kominek and Black, 2003], which was a typical amount of data used for voice transformation at the time. Then voice transformation models were trained in both directions for each of the speaker pairs (1 male pair and 1 female pair for each group for a total

[1] The sections of this chapter on subjective measures are adapted from a conference paper [Toth and Black, 2006].

of 4 pairs). Voice transformation was performed by scaling pitch estimates, using a Gaussian Mixture Model mapping to transform mel-cepstral coefficients, and using a MLSA filter [Imai, 1983] for synthesis as described in [Toda, 2003].

For each speaker pair, a pair comparison evaluation with 10 trials was constructed. The utterances in each pair had different text to avoid confusion from the unmodified portions of source speaker prosody, such as power, that were carried over to the transformed speech. Some trials consisted of recordings from different speakers, some consisted of transformed speech in different directions between the speakers, and some consisted of a recording and transformed speech. The original recordings were analyzed and resynthesized using the same MLSA filter technique [Imai, 1983] employed by the voice transformation process, in order to minimize differences perceived from artifacts due to the vocoding process used during transformation. Listeners were asked to rate the similarity of the speakers in each trial on a scale from 1 to 5, where 1 meant the speakers were very similar and 5 meant the speakers were very different. How the listeners were to judge speaker similarity and difference was left to them. In total, 10 listeners (5 from each group) listened to 40 utterance pairs (10 utterance pairs for each of 4 speaker pairs). With this setup we were able to collect data to investigate whether knowing the speakers made a difference in the judgment of speaker recognizability for voice transformation.

6.5 Data Analysis

One thing we wanted to know immediately was whether the voice transformation was "successful." One measure of this was whether the transformed speech was consistently judged as being more similar to the target speaker than the source speaker. This, indeed, was the case when considering the average similarity scores for each speaker pair across all listeners. These averages are shown in Figure 6.1, where "s1" stands for the first speaker in each pair, "s2" stands for the second speaker in each pair, "s1→s2" stands for transformed speech with the first speaker as the source and the second speaker as the target, and "s2→s1" stands for transformed speech with the second speaker as the source and the first speaker as the target. The scores comparing the target speakers with the transformed speech (s2,s1→s2 and s1,s2→s1) were lower, and thus more similar, than the scores comparing the source speakers with the transformed speech (s1,s1→s2 and s2,s2→s1).

Looking at the bars in Figure 6.1, a few more trends become apparent. Moving from the leftmost group of bars to the rightmost group, the bars for each speaker pair tend to get higher, showing greater differences in the compared speech. It appears that as the speakers are themselves judged further apart, the transformed speech is also judged as

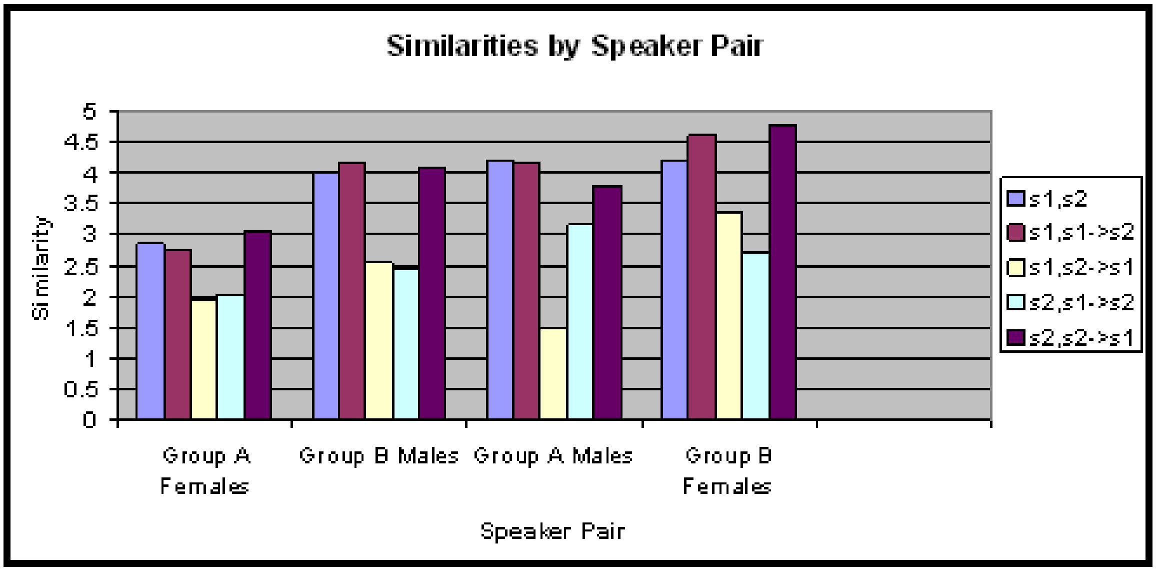


Figure 6.1: Similarities by Speaker Pair

being further from the speakers. The Group A male speakers stand out as having the only exceptions to this general rule. Interestingly, there is a strong asymmetry with the Group A male speakers. The bar comparing the transformation s2→s1 with its target speaker, s1, is much shorter than the bar comparing the transformation s1→s2 with its target speaker, s2. This suggests that the transformation from speaker s2 to s1 was much more successful than the transformation from speaker s1 to s2.

The next question was whether knowing the speakers made a difference. A breakdown of the results according to whether the listeners knew the speakers is given in Figure 6.2. Not only did the same general trend appear, where the transformed speech was judged as being more similar to the target speech than the source speech, but the scores for each type of compared speech were very close regardless of whether the listeners knew the speakers.

6.6 Transformation Triangle Diagrams

As we looked at numerous graphs similar to the ones in Figure 6.1 and Figure 6.2, we realized that we wanted a better way to summarize multiple bars in the graphs and show how their values were related to each other. This led us to create **Transformation Triangle Diagrams (TTDs)** for each speaker pair. Some examples of these are in Figure 6.3, Figure 6.4, Figure 6.5, and Figure 6.6. TTDs can be interpreted as follows:

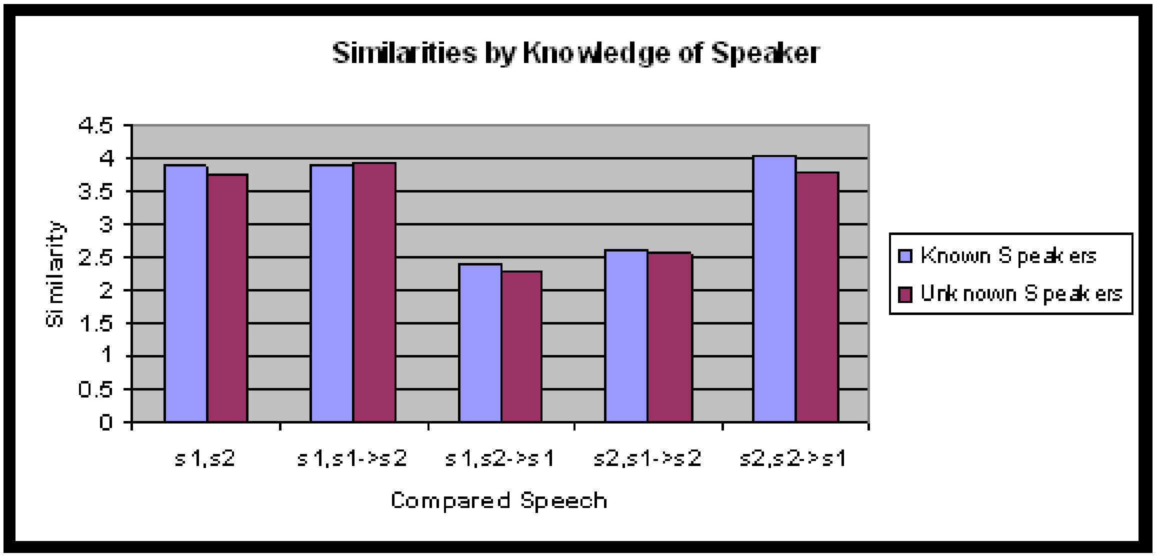


Figure 6.2: Similarities by Knowledge of Speaker

- The numbers in the diagrams are calculated by subtracting 1 from the similarity scores to compute 0-based similarity “distances” where 0 is most similar and 4 is most different.

- The distance between speech from the two speakers in a pair is represented by a horizontal line, with the names of the speakers listed at either end.

- Each diagram is composed of two directed triangles. The upper triangle represents comparisons made using the left speaker in the TTD as the source for voice transformation and the right speaker as the target. The lower triangle represents comparisons made using the right speaker as the source for voice transformation and the left speaker as the target. The arrows serve as reminders for the directions of the transformations.

- The vertices that are off the horizontal baseline represent transformed speech, and the remaining triangle edges represent the distances from the speakers’ speech to the transformed speech. For example, in the first TTD in Figure 6.3, the distance between speaker a1 and speech transformed from a1 to a2 is 1.9, the distance between speech transformed from a1 to a2 and speaker a2 is 0.7, the distance between speaker a2 and speech transformed from a2 to a1 is 2.1, and the distance between speech transformed from a2 to a1 and speaker a1 is 0.5

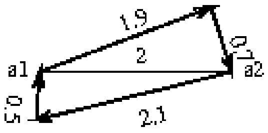


Figure 6.3: Transformation Triangle Diagram Example 1

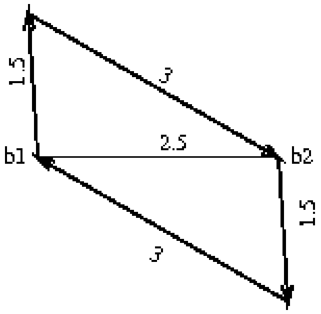


Figure 6.4: Transformation Triangle Diagram Example 2

- It should be noted that TTDs make no attempt to compare transformed speech using one speaker as the source with transformed speech using the other speaker as the source.

A few examples of TTDs are given in Figure 6.3, Figure 6.4, Figure 6.5, and Figure 6.6. Figure 6.3 represents a pair of speakers called a1 and a2, where both transformations were mostly successful in that the transformed speech was considerably closer to the targets than the sources in both cases.

Figure 6.4 represents a pair of speakers called b1 and b2, where both transformations were fairly unsuccessful in that the transformed speech was closer to the source than the target. As transformation becomes more successful, the TTDs tend to skew so the upper triangle is crushed to the right and the lower triangle is crushed to the left.

However, distance from a vertex representing transformed speech to the horizontal baseline can make a difference as well. In Figure 6.5 representing speakers c1 and c2 and in Figure 6.6 representing speakers d1 and d2, the vertices representing the transformed speech would project to the same location on the horizontal baselines, but the transformations between c1 and c2 were more successful than the ones between d1 and d2 because the transformed speech is closer to the targets. One additional point is that the length of the horizontal baselines vary according to the similarity of the speakers. The more similar the speakers are, the narrower the baseline is.

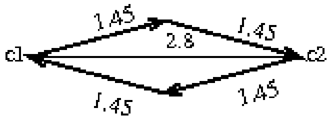


Figure 6.5: Transformation Triangle Diagram Example 3

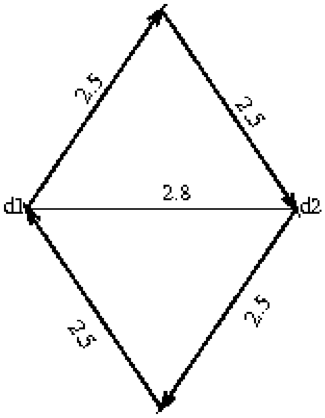


Figure 6.6: Transformation Triangle Diagram Example 4

In the ideal case, both transformations would coincide with their targets, and the TTD would collapse to a horizontal line with arrowheads pointing outward at each end. In a case where the transformation was completely unsuccessful and the transformed speech sounded like the source voice, the TTD would again collapse to a horizontal line, but there would be inward pointing arrows as well.

It is important to note that the distances in these diagrams may not actually be distances in a Euclidean sense, and it may not be possible to construct triangles for some combinations of scores if the lengths of the edges do not satisfy the triangle inequality. One pathological case would be when the horizontal bar is longer than the sum of the other two sides of a triangle. That would mean that the distance between the source and target speakers is actually greater than the combined distances of the transformed speech to both the source and target speakers. The other pathological case would be when the distance from the transformed speech to one of the speakers was greater than the sum of the distance from the transformed speech to the other speaker plus the distance between the two speakers themselves. In such a case, it would also be impossible to construct a triangle. However, it should be noted that for all the examples we tried based on our data, we were able to construct triangles

TTDs are not the first attempt to try to represent distances between speech in voice transformation. Others have used Multi-Dimensional Scaling (MDS) techniques to accomplish this [Abe et al., 1988]. In MDS, distances are calculated among multiple quantities in a multi-dimensional space, and the results are projected onto a plane for comparison. Although MDS is an interesting and useful technique for analyzing data, we find that TTDs are a compact, simpler-to-understand way of depicting the specific relationships we are trying to compare in voice transformation.

6.7 Evaluating Voice Transformation with TTDs

The TTDs for results from our listening experiment broken down by speaker pair are in Figure 6.7. These results correspond to the four speaker pairs from the graph in Figure 6.1. Looking at these TTDs, a number of things become readily apparent. First of all, the transformations were mostly successful in the sense that the triangles are skewed so the transformed speech is closer to the target speech than the source speech in each case. Another point is that the speakers in the first pair were considered much more similar than the others based on the widths of the diagrams. One interesting thing that appears in the third pair is that the transformation from **ref** to **ehn** is much more successful than the transformation from **ehn** to **ref**, as shown by the asymmetry in the diagram. This is another visual depiction of the same asymmetry mentioned earlier in the section on Data Analysis.

6.8 Subjective Measures Discussion

In our listening experiment, we found that whether the listeners knew the speakers did not appear to significantly affect how they judged speaker similarity. This knowledge will guide us in designing further experiments of this nature because we will not be concerned with finding listeners who either know or don't know the speakers. We have also created a new type of diagram called a **Transformation Triangle Diagram (TDD)** that was useful in representing certain relationships in a compact, understandable manner. Future work will involve investigating further methods of visualizing voice transformation results. While this paper investigates the area of speaker recognizability, there are other areas of voice transformation evaluation, such as intelligibility and naturalness, where different forms of analysis may be necessary.

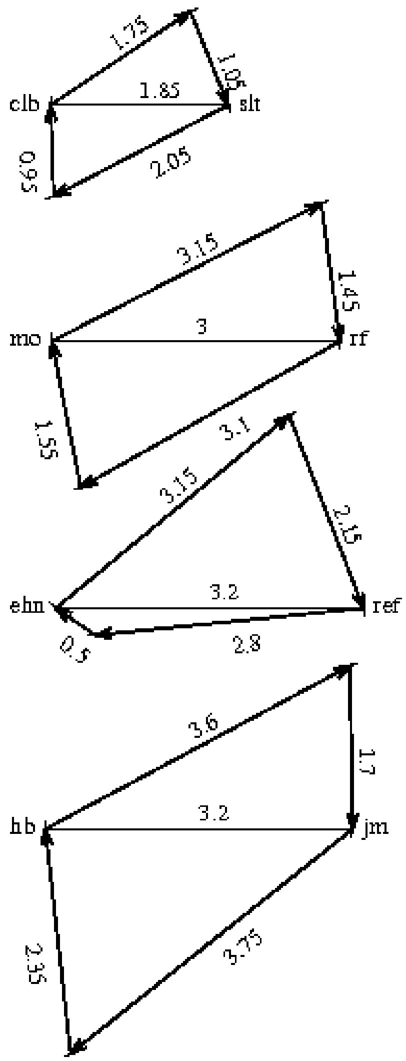


Figure 6.7: Transformation Triangle Diagrams by Speaker Pair

6.9 Automatic Speaker Identification for Measuring VT

We also explored new techniques for objective VT measures. Numerous publications mention the possibility of using VT to fool speaker verification, but only a few give results [Pellom and Hansen, 1999] [Masuko et al., 2000] [Perrot et al., 2005]. We performed some basic experiments in this area and then investigated novel approaches of using **Speaker IDentification (SID)** systems to measure the quality of the identity of transformed speech. SID systems are typically composed of **speaker models** which score the closeness of test speech qualities to a particular speaker's characteristics. Many traditional speaker identification tasks are essentially classification tasks, and the test speech is scored by numerous speaker models before deciding which one is the best match. We did indeed perform such experiments with transformed speech under **closed-set** conditions, where test speech had to be classified as coming from one of a given set of speakers and there was no option to reject the speech as being from none of them. With transformed speech, however, we are primarily interested in how well it matches the target speaker's characteristics, and competing transformation and synthesis techniques can be compared against a model based on the target speaker's speech. Instead of scoring one target speaker according to many speaker models, as in typical SID tasks, multiple types of target speech with the same target identity are scored against a single model in this approach.

When evaluating transformed speech, it may also be desirable to know whether its identity is closer to the target speaker than the source speaker. In this case, it would be appropriate to consider scores from two speaker models, one for the source speaker and one for the target speaker. This, however, would not help when comparing the transformed speech with types of synthetic speech where data from only one speaker is used to construct the synthetic voice.

The question of measuring speaker identity is discussed in Section 6.10. A description of the speaker identification systems used in this work is in Section 6.11. Our first experiments to test the general properties of the speaker identification systems on transformed speech are described in Section 6.13. From there, we proceed to using SID systems to score VT and compare it with speech synthesis approaches. The speech synthesizers we consider and their resulting scores are described in Section 6.14.1 and Section 6.14.2. The results of all these experiments are discussed generally in Section 6.15.[2]

[2] The work in this chapter involving SID systems was performed in collaboration with Qin Jin, Tanja Schultz, and Alan Black. Qin ran the SID systems to score various speech data I produced. Some figures in this chapter were produced by Qin Jin and Tanja Schultz.

6.10 Measuring Identity

The speaker identity of transformed speech is typically measured through subjective listening tests involving human judgments, but these tests are costly and time consuming. Furthermore, the two most popular identity tests used for VT, ABX tests and pair comparison tests, are difficult to interpret [Kain, 2001] [Abe et al., 1988] [Sündermann et al., 2006] [Toth and Black, 2006]. For these reasons, we seek an automatic measure of identity that can be performed without the costs of human experiments.

One possibility is to use scores from **Speaker Identification (SID)** systems typically used for recognizing the identity of an unknown speaker. These scores represent distances between speakers.

6.11 Speaker Identification Systems

The following experiments use two SID systems, which were both created by Qin Jin [Jin et al., 2008]. One is a GMM-based SID system that is an example of the most prevalent method of performing SID [Reynolds and Rose, 1995] [Reynolds et al., 2000]. The other is a newer, competitive approach called Phonetic SID [Jin et al., 2002].

6.11.1 GMM-based SID System

In the GMM-based SID system, a **speaker model** is created for each speaker by training a GMM on spectral features. More specifically, 13-dimensional mel-cepstral features are extracted from a training set of speech, and the parameters for a GMM with 256 components are estimated using the EM algorithm. The resulting GMM for a speaker is the speaker model. Test speech is scored by extracting mel-cepstral features from it and measuring its likelihood based on a speaker model.

6.11.2 Phonetic SID System

In the phonetic SID system, a speaker model is created based on phone recognizers built in multiple languages using external data. In our experiments, the phone recognizers were built for 12 different non-English languages using data from the GlobalPhone project [Schultz and Waibel, 1997]. A speaker model was created by learning a **Language Specific Phonetic Model (LSPM)** for each of the 12 languages for a speaker. An LSPM was constructed by taking one of the phone recognizers trained on GlobalPhone data for a specific language, using it to decode non-English phones on the English SID training data for a particular speaker, and building an **n-gram language model** based on the recognition results.

Test set speech was scored by decoding it with the 12 GlobalPhone phone recognizers to produce phone sequences for the 12 different languages, measuring the likelihoods of these decoded phone sequences according to the corresponding LSPM trained for the specific speaker, and combining the likelihood results from the 12 LSPMs for that speaker. The current technique of combining the LSPM results for a single speaker is to sum the LSPM likelihoods.

Although it may seem a bit peculiar that the phone recognizers were trained on languages other than the one in which the test data was recorded, the Phonetic SID System has been shown to perform well on standard SID tasks [Jin et al., 2002]. The underlying intuition is that the multi-lingual phonetic units still exist in a space where useful distinctions about the test language phones can be captured and that their distributions characterize different speakers.

6.12 Data

The data used in these experiments came from the LDC CSR-I (WSJ0) Complete corpus [Linguistic Data Consortium, 1993]. We extracted features from the recorded speech and manually corrected the provided transcripts. The speakers spoke General US English, and the recordings had little noise. After removing duplicate sentences from the corpus, we chose the 24 speakers who had at least 55 remaining sentences. For each speaker, 50 sentences were used for training and 5 for testing.

The VT trials required additional source speakers. We used synthesized utterances made with the kal-diphone voice [Lenzo and Black, 2000] from the Festival Speech Synthesis System distribution [Black and Taylor, 1997b] and new recordings from a General US English speaker. We chose the kal-diphone voice for its consistent quality and the fact that it is freely available, which makes it easier for others to duplicate our experiments. The utterances for all 24 speakers were synthesized, but due to time constraints, we only created new recordings of the utterances for 8 of them.

6.13 Speaker Identity Experiments

6.13.1 Recorded Speech

The first experiments we performed were for confirming that our SID systems performed reasonably in terms of recorded speech. For both SID systems, speaker models were created based on the training set recordings from all 24 speakers. Then the test sentences were evaluated based on these models. For both SID systems and for all speakers, the models corresponding to the actual speakers gave the highest likelihood scores for the test sentences. This confirmed that both the GMM-based and Phonetic SID systems were able to successfully distinguish among the 24 speakers.

6.13.2 Single-Model Experiments

One basic question we had was how transformed speech would be classified according to speaker models based on recorded speech. We called these experiments **single-model experiments** because, for a SID system, there was one speaker model per original speaker. For each of the 24 recorded speakers, we used our baseline VT system to construct a synthetic version of the test set by using kal-diphone synthetic speech for the source speaker and recorded speech for a target speaker. Using synthetic speech for the source speaker is convenient as it does not require recording additional sentences. Also, it matches the scenario of one of the desirable applications for voice transformation: using a small amount of additional recorded data to modify a synthetic voice based on a larger amount of data.

The two SID systems gave different results for the transformed speech. The confusion diagram for the GMM-based SID system is in Figure 6.8, and the confusion diagram for the Phonetic SID system is in Figure 6.9.[3] In both diagrams, the horizontal axis represents the target of the transformed speech. “V01” through “V24” represent transformed speech with target speakers 1 through 24. In both diagrams, the vertical axis represents the speaker from whose utterances the speaker model was created. “S01” through “S24” represent recorded speech from speakers 1 through 24.

For the GMM-based system, the transformed speech was always judged to be most like the actual speech of the corresponding target speaker. In a sense, it can be said that the VT system fooled the GMM-based SID under **closed-set** conditions. On the other hand, the Phonetic SID system, except for one case, always judged the transformed speech to be most like one of two recorded speakers, regardless of the VT target. In a sense, it can

[3] Both diagrams were created by Tanja Schultz for our ICASSP 2008 paper [Jin et al., 2008].

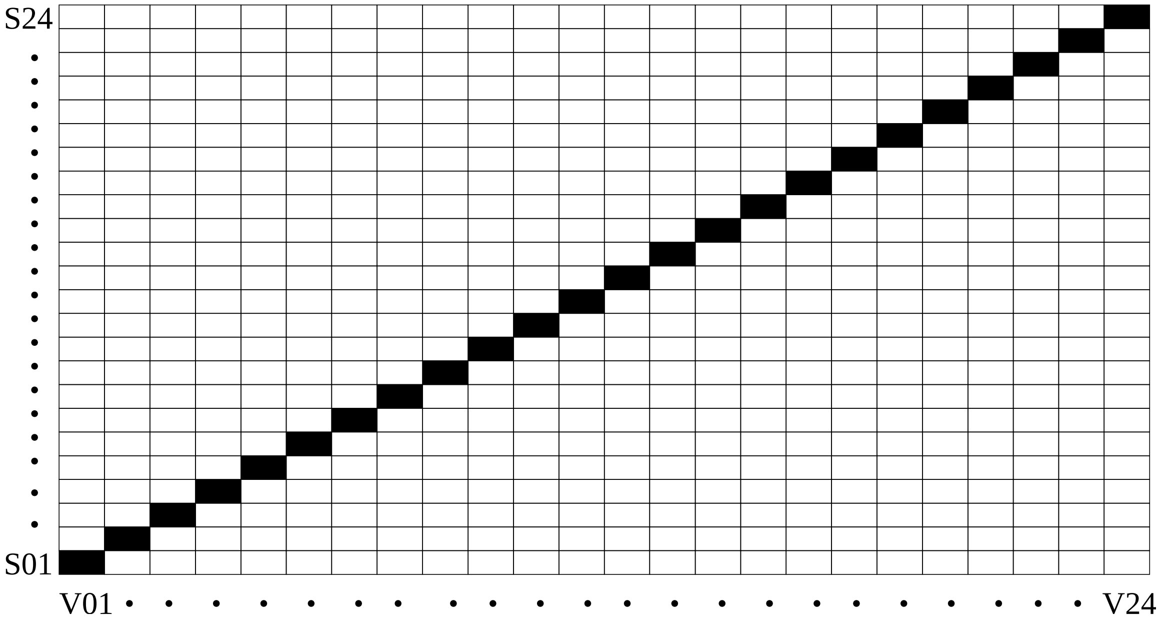


Figure 6.8: Confusion Matrix for Single-Model Experiments with Transformed Speech and GMM-based SID

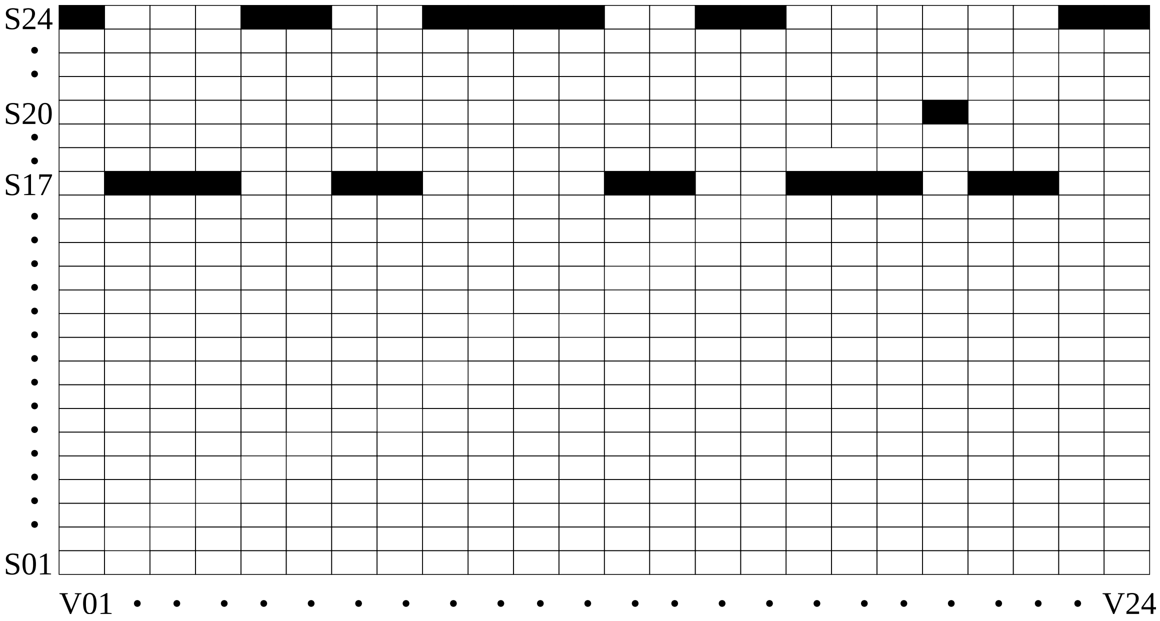


Figure 6.9: Confusion Matrix for Single-Model Experiments with Transformed Speech and Phonetic SID

be said that the VT system failed to fool the Phonetic SID system. Further investigation showed that utterances created using the kal-diphone synthetic voice were judged by the Phonetic SID system to be closest to the same two recorded speakers as the transformed speech. This suggests that from the perspective of the Phonetic SID system, the transformation was unsuccessful as the identity was judged to be like the source speaker instead of the target speaker.

6.13.3 Dual-Model Experiments

Another question we had was what would happen if we extended our experiments to include speaker models based on transformed speech as well. We called these experiments **dual-model experiments** because, for each original speaker there were two speaker models: one based on recorded speech and another based on transformed speech.

For the GMM-based system, transformed speech was always judged to be most like the model based on transformed speech with the same target, but the model based on speech recorded from the target speaker almost always was a top-5 hypothesis. For the Phonetic based system, transformed speech was always judged to be most like the model based on transformed speech with the same target, and the model based on the speech recorded from the target speaker was never a top-5 hypothesis. Though neither SID system appeared to be fooled by the transformed speech, consideration of the top-5 hypotheses suggests that the Phonetic SID system is more robust when attacked with transformed speech.

6.14 Experiments With Measuring Identity

We performed experiments to score how well the transformed speech matched the target speakers and compare these results with synthetic speech created from the target speaker data without using source speaker data. Two different types of synthesizers, CLUSTERGEN statistical parametric synthesis [Black, 2006] and cluster unit selection [Black and Taylor, 1997a], were compared to voice transformation. The normalized scores for the GMM-based SID system are shown in Figure 6.10, and the normalized scores for the Phonetic SID system are shown in Figure 6.11.[4] The synthesizers and results are discussed in the following sections.

[4] These two figures were created by Qin Jin with my help.

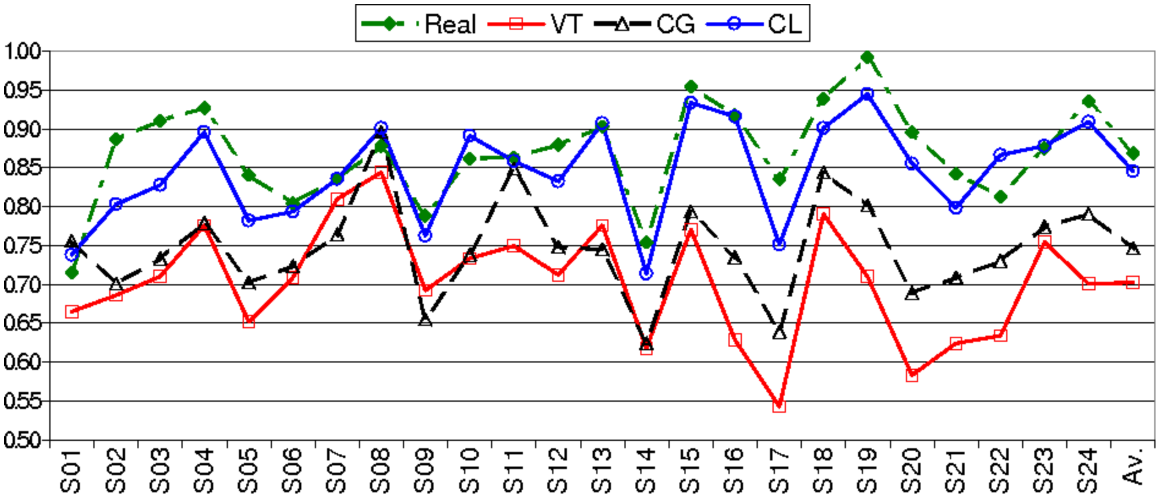


Figure 6.10: Normalized GMM-based SID Scores for Various Types of Speech. Smaller values represent being closer to the target speaker and are better. “Real” is recorded speech, “VT” is voice transformed speech, “CG” is CLUSTERGEN synthetic speech, and “CL” is cluster unit selection synthetic speech.

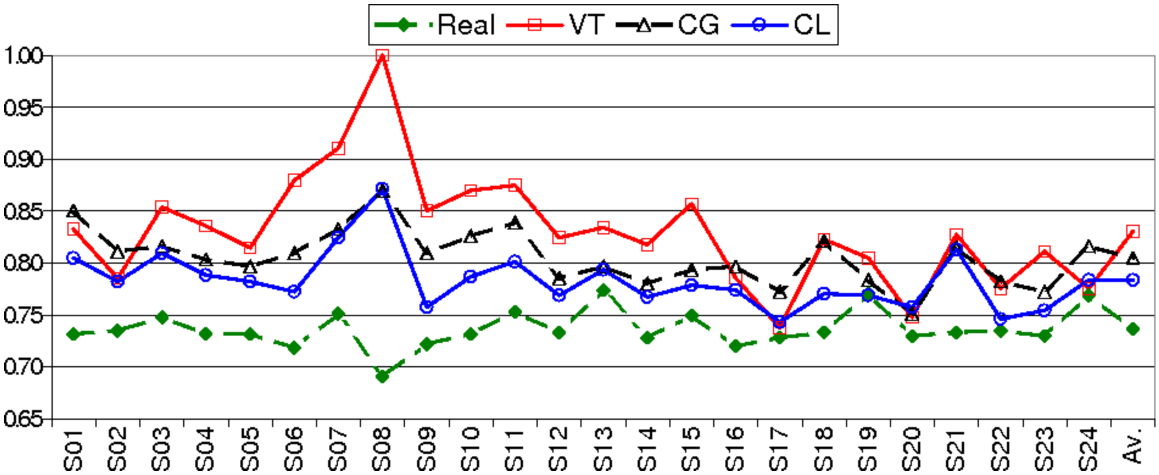


Figure 6.11: Normalized Phonetic SID Scores for Various Types of Speech. Smaller values represent being closer to the target speaker and are better. “Real” is recorded speech, “VT” is voice transformed speech, “CG” is CLUSTERGEN synthetic speech, and “CL” is cluster unit selection synthetic speech.

6.14.1 CLUSTERGEN Synthesizer

CLUSTERGEN [Black, 2006] is a statistical parametric speech synthesis technique [Black et al., 2007], similar to the HMM-generation synthesis techniques found in HTS [Zen et al., 2007]. Its advantage is that it can work better than many other synthesis techniques with small amounts of data. Another potential advantage is that it makes predictions in phonetic contexts, and the incorporation of phonetic level information may be helpful when identity is evaluated by Phonetic SID. One disadvantage is that the final production of the audio signal can sound unnatural due to signal processing artifacts.

CLUSTERGEN voices were created with training data from the 24 WSJ speakers, and were evaluated with the GMM-based and Phonetic SID systems. for the GMM-based system, for all synthetic CLUSTERGEN voices, the speaker model trained on recordings from the corresponding WSJ speaker gave the best score. For the Phonetic SID system, in 11 cases, the speaker model trained on recordings from the corresponding WSJ speaker gave the best score, which was an improvement when compared to the transformed speech.

6.14.2 Cluster Unit Selection Synthesizer

Cluster unit selection [Black and Taylor, 1997a] is a concatenative speech synthesis technique. It splits training files into smaller segments which are reassembled in different orders to form synthetic utterances. One advantage of this technique is that the output segments are actual speech and do not suffer from unnatural signal processing artifacts, though the places where the segments are joined can sound unnatural. Also, the audio segments are longer than individual frames, and this may be an advantage over VT in Phonetic SID evaluation. One disadvantage is that it tends to require more data than CLUSTERGEN. In fact, one of the utterances in the test set could not be synthesized by the Cluster unit selection approach because there was insufficient phonetic coverage in the training set.

Cluster unit selection synthetic voices were created with the training data from the 24 WSJ speakers and evaluated with both the GMM-based and Phonetic SID systems. for the GMM-based and Phonetic SID systems. for the GMM-based SID systems, for each synthetic voice test, the speaker model trained on recordings from the corresponding WSJ speaker was the closest, and in one case (speaker S02) was even closer to the synthetic voice than the speaker model trained on the synthetic training data. For the Phonetic SID system, in 20 cases, the speaker model trained on recordings from the corresponding WSJ speaker gave the best score, and in two cases was even closer to the synthetic voice than the model trained on the synthetic training data. Thus cluster unit selection performed better in terms of identity then CLUSTERGEN with respect to the two SID systems.

6.15 Objective Measure Discussion

Returning to the original question of what SID systems tell us about the identity of VT and synthetic speech, we find a number of answers. From the perspective of a GMM-based SID system, VT and two types of synthetic speech are closer in identity to the real speech SID models than additional real speech from the same speakers. It appears that the models used in VT and the synthetic voices capture characteristics of the training data so well, from the perspective of the GMM-based SID system, that the variation between speech generated from the model is even less than the variation within the recorded speech from a single speaker, going from training set to test set. Furthermore, CLUSTERGEN synthesis performed better than cluster unit selection, and both scored worse than VT, but better than real speech. However, from the perspective of the Phonetic SID system, the rankings with real speech being closer in identity to the real speech SID models than the two types of synthetic and VT speech tells us the opposite.

How can it be that two SID systems which give completely consistent results on classifying a set of recorded speakers give such completely different results when ranking VT and speech synthesis? The answer appears to be that speaker identity is a complicated quality that is based on multiple components, and that the GMM-based and Phonetic SID systems emphasize different aspects. The GMM-based SID system focuses on statistics from short frames of speech and does not consider longer audio units. In contrast, the Phonetic SID system focuses on statistics from sequences of phones, which in themselves are typically longer than frames. What SID systems tell us about VT and speech synthesis is that they are deficient in terms of the longer range statistics that are processed in Phonetic SID systems.

As processes for creating synthetic speech improve, SID systems will need to improve to defend against impostor attacks based on them. These new SID systems, in turn, will provided new metrics for evaluating synthetic speech. Thus the two types of systems can be used to improve each other. However, the SID systems must correlate with an independent identity judgment, so the competition does not optimize an unrelated quantity.

In addition to identity, naturalness and intelligibility are important to synthetic speech. In informal listening tests, we found the synthetic speech examples in these experiments unnatural and difficult to understand, yet these examples fared well against SID systems. When speaker data is limited, there appears to be a trade-off between speaker identity and naturalness [Fernandez et al., 2006]. Given the starting points here, it appears that either naturalness and intelligibility of speech synthesis need to be improved, which may be difficult or impossible using a small training set, or the additional external data used in VT that can help with naturalness and intelligibility needs to be used without interfering with identity.

Chapter 7

Conclusions

7.1 Summary

This thesis explored the possibility of using articulatory position data to improve voice transformation. Chapter 2 presented background material on speech models, which are necessary components of voice transformation systems. They are discussed, both in their historical context and with respect to which ones are used most commonly in voice transformation. In Chapter 3, voice transformation was described, and the line of work that led to what is currently the most prominent voice transformation technique was explored. In addition, the baseline voice transformation system used in numerous experiments in later chapters was explained in detail. Chapter 4 showed the use of EMA data to improve the baseline voice transformation system described in Chapter 3. The most straightforward modifications to the system did not help. Finally, after attempting a combination of changes, both to the voice transformation process and the use of the data, there was a small positive improvement for one direction of voice transformation. Chapter 5 investigated the question of whether articulatory position data from a speaker could be used with another speaker. This is important because articulatory position data is difficult to collect and in most cases probably will not be available for a specific speaker. Experiments showed that pseudo-articulatory data, produced by mappings from MCEP features, could be used to improve some phonetic feature predictions and some voice transformations. Finally, in Chapter 6, the question of how to evaluate the quality of voice transformation was explored. An experiment showed that the results of a subjective listening test were not affected by whether the listeners knew the speakers whose voices were transformed. Also, a new method of visualizing results from a subjective pair comparison test was presented. New objective metrics, based on automatic speaker identification systems were also investigated. They demonstrated that GMM-mapping based voice transformation systems have specific strengths and weaknesses in comparison to various other synthesis techniques.

7.2 Contributions

- An investigation of extending GMM-mapping based voice transformation with articulatory position data
- Creation of techniques that allow the use of articulatory position data from one speaker with another
- Demonstration that cross-speaker articulatory position prediction techniques can be used to improve phonetic feature prediction and voice transformation
- Showed that it does not matter whether listeners in a pair comparison test for voice transformation know the speakers
- Created a new way of visualizing pair comparison tests for voice transformation
- Produced a novel way of measuring the identity of transformed speech using automatic speaker identification systems

7.3 Future Work

There is still much room for improvement in voice transformation, and in the use of articulatory position data. Some of the future challenges for voice transformation are to incorporate more higher-level features, such as a more sophisticated prosodic model and word choice. Some of the future challenges involving articulatory position data involve making it easier to collect more consistent and more complete data. As for the evaluation of voice transformation, there is still a desire to create objective measures that are more consistent with subjective measures. Such objective measures would greatly simplify the evaluation of voice transformation and might also be useful measures during the transformation process itself.

Bibliography

M. Abe. A segment-based approach to voice conversion. In *ICASSP*, volume 2, pages 765–768, 1991. 3.1

M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. In *Proceedings of ICASSP 88*, pages 655–658, Tokyo, 1988. 3.3.1, 3.5.2, 6.6, 6.10

M. Abe, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *J. Acoust. Soc. Jpn. (E)*, 11:71–76, 1990a. 3.3

M. Abe, K. Shikano, and H. Kuwabara. Cross-language voice conversion. In *ICASSP*, pages 345–348. IEEE, 1990b. 3.2

M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Statistical analysis of bilingual speaker's speech for cross-language voice conversion. *The Journal of the Acoustical Society of America*, 90(1):76–82, 1991. 3.2

B. S. Atal and J. R. Remde. A new model of lpc excitation for producing natural-sounding speech at low bit rates. In *Proceedings of ICASSP 82*, pages 614–617, Paris, France, April 1982. 2.2.2

B. S. Atal and M. R. Schroeder. Predictive coding of speech signals. In *Conf. Commun. and Process.*, pages 360–361, 1967. 2.2

Kleijn W. B. and K. K. Paliwal, editors. *Speech Coding and Synthesis*. Elsevier, 1995. 2.3.4

A. Black. CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling. In *Interspeech 2006 - ICSLP*, 2006. 6.14, 6.14.1

A. Black and K. Lenzo. Building voices in the Festival speech synthesis system. http://festvox.org/bsv/, 2000. 3.4

A. Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Eurospeech97*, volume 2, pages 601–604, Rhodes, Greece, 1997a. 6.14, 6.14.2

A. Black and P. Taylor. The Festival Speech Synthesis System: system documentation. Technical Report HCRC/TR-83, Human Communication Research Centre, University of Edinburgh, Scotland, UK, Jan. 1997b. Available at http://www.cstr.ed.ac.uk/projects/festival/. 6.12

A. Black, H. Zen, and K. Tokuda. Statistical parametric speech synthesis. In *ICASSP*, pages 1229–1232, 2007. 6.14.1

B. P. Bogert, M. J. Healy, and J. W. Tukey. *The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking*, chapter 15, pages 209–243. Wiley, New York, 1963. 2.4

O. Cappé, J. Laroche, and E. Moulines. Regularized estimation of cepstrum envelope from discrete frequency points. In *IEEE ASSP Workshop on App. of Sig. Proc. to Audio and Acoust.*, Mohonk, October 1995. 3.3.2

Carnegie Mellon University. SphinxTrain: building acoustic models for CMU Sphinx. http://www.speech.cs.cmu.edu/SphinxTrain/, 2001. 5.3.3

D. G. Childers, D. P. Skinner, and R. C. Kemerait. The cepstrum: A guide to processing. *Proceedings of the IEEE*, 65(10):1428–1443, October 1977. 2.4

D. G. Childers, B. Yegnanarayana, and Ke Wu. Voice conversion: Factors responsible for quality. In *ICASSP 1985*, pages 748–751, 1985. 3.3

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (Series B:1–38, 1977. 3.3.2, 3.3.2

H. Dudley. Remaking speech. *The Journal of the Acoustical Society of America*, 11(2): 169–177, 1939. 2.3, 2.3.2, 2.3.2

D. Erro, T. Polyakova, and A. Moreno. On combining statistical methods and frequency warping for high-quality voice conversion. In *Proc. ICASSP2008*, Las Vegas, NV, USA, Mar. 2008. 1.3

R. Fernandez, R. Bakis, E. Eide, W. Hamza, J. Pitrelli, and M. Picheny. The 2006 TC-STAR evaluation of the IBM text-to-speech synthesis system. In *TC-STAR Workshop on Speech-to-Speech Translation*, 2006. 6.15

J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell System Technical Journal*, 45: 1493–1509, 1966. 2.3.3

J. Frankel, M. Wester, and S. King. Articulatory feature recognition using dynamic bayesian networks. In *Proceedings ICSLP2004*, Oct. 2004. 5.2, 5.3.1

B. Gold and C. M. Rader. The channel vocoder. *IEEE Transactions on Audio and Electroacoustics*, AU-15(4):148–161, December 1967. 3, 4

D. W. Griffin and J. S. Lim. Multiband excitation vocoder. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-36(8):1223–1235, 1988. 2.3.5

M. Hart. Project Gutenberg. http://promo.net/pg/, 2000. 5.4.1

W. Hess. *Pitch Detection in Speech Signals: Algorithms and Devices*. Springer Verlag, 1983. 2.2.2

S. Hiroya and M. Honda. Acoustic-to-articulatory inverse mapping using an HMM-based speech production model. In *ICSLP2002*, Denver, CO., 2002. 5.2

R. R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49, 1976. 5.3.3

Hull. *Gen. Elec. Rev.*, 32:397, 1929. 2.3.2

Hull. *Physics*, 4:75, 1933. 2.3.2

S. Imai. Cepstral analysis synthesis on the mel frequency scale. In *Proceedings of ICASSP 83*, pages 93–96, 1983. 2.4.3, 6.4

F. Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech, and Signal Processing*, 23(1):67–72, Feb. 1975. 3.3.2, 3.4.1, 5.4.2

Q. Jin, T. Schultz, and A. Waibel. Phonetic speaker identification. In *ICSLP2002*, pages 1345–1348, Denver, CO., 2002. 6.11, 6.11.2

Q. Jin, A. Toth, A. Black, and T. Schultz. Is voice transformation a threat to speaker identification? In *ICASSP 2008*, 2008. 1.6, 3.2, 6.11, 3

Q. Jin, A. Toth, T. Schultz, and A. Black. Voice convergin: Speaker de-identification by voice transformation. In *Proc. ICASSP 2009*, 2009. 3.2

A. Kain. *High Resolution Voice Transformation*. PhD thesis, OGI School of Science and Engineering, OHSU, 2001. 3.1, 3.2, 3.3, 3.3.2, 3.3.2, 3.5.2, 6.2, 6.10

A. Kain, X. Niu, J. Hosom, Q. Miao, and J. van Santen. Formant re-synthesis of dysarthric speech. In *5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, 2004. 3.2

H Kawahara, I. Masuda-Katsuse, and A. de Cheveigné. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: possible role of a repetitive structure in sounds. *Speech Communication*, 27(3-4):187–207, 1999. 3.3.2

J. Kominek and A. Black. The CMU ARCTIC speech databases for speech synthesis research. Technical Report CMU-LTI-03-177 http://festvox.org/cmu_arctic/, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, 2003. 5.7.1, 6.4

K. Lenzo and A. Black. Diphone collection and synthesis. In *ICSLP2000*, volume III, pages 306–309, Beijing, China., 2000. 6.12

K. Lenzo and O. Fujimura. Microbeam.org. http://microbeam.org, 2001. 1.3

J. S. Lim and A. V. Oppenheim. *Advanced Topics in Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1988. 2

Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95, Jan. 1980. 3.4.1

Linguistic Data Consortium. CSR-I (WSJ0) complete. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S6A, 1993. 6.12

J. D. Markel and A. H. Gray Jr. *Linear Prediction of Speech*. Springer-Verlag, Berlin Heidelberg New York, 1976. 2.2

K. Markov, J. Dang, Y. Iizuka, and S. Nakamura. Hybrid HMM/BN ASR system integrating spectrum and articulatory features. In *Eurospeech03*, Geneva, Switzerland, 2003. 5.2

K. Markov, S. Nakamura, and J. Dang. Integration of articulatory dynamic parameters in HMM/BN based speech recognition system. In *Proc. ICSLP2004*, Oct. 2004. 5.2

T. Masuko, K. Tokuda, and T. Kobayashi. Imposture using synthetic speech against speaker verification based on spectrum and pitch. In *Proceedings ICSLP2000*, 2000. 3.2, 6.9

R.J. McAulay and T.F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35: 744–754, Aug. 1986. 2.3, 2.3.4

R.J. McAulay and T.F. Quatieri. Computationally efficient sine-wave synthesis and its application to sinusoid transform coding. In *Proceedings of ICASSP 88*, pages 370–373, Tokyo, Apr. 1988. 2.3.4

L. Mesbahi, V. Barreaud, and O. Boeffard. GMM-based speech transformation systems under data reduction. In *ISCA SSW6*, 2007. 3.2

F. Metze and A. Waibel. A flexible stream architecture for ASR using articulatory features. In *ICSLP2002*, Denver, CO., 2002. 5.2, 5.3.1, 5.3.3, 5.8

H. Mizuno and M. Abe. Voice conversion algorithm based on piecewise linear conversion rules of formant frequency and spectrum tilt. *Speech Communication*, 16(2):153–164, 1995. 3.2

A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, Upper Saddle River, New Jersey 07458, 1999. Second Edition. 2

M. Ostendorf. Moving beyond the 'beads-on-a-string' model of speech. In *Proc. ASRU 99*, 1999. 5.2

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philisophical Magazine*, 2:559–572, 1901. 4.5.4

B. Pellom and J. Hansen. An experimental study of speaker verification sensitivity to computer voice-altered imposters. In *ICASSP-99*, pages 837–840, Phoenix, Arizona, 1999. 3.2, 6.9

P. Perrot, G. Aversano, R. Blouet, M. Charbit, and G. Chollet. Voice forgery using ALISP: Indexation in a client memory. In *ICASSP*, 2005. 6.9

M. Portnoff. Short-time fourier analysis of sampled speech. *IEEE Trans. Acoust., Speech, and Signal Proc.*, ASSP-29(3):364–373, 1981. 2.3.3

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK., 2nd edition, 1992. 2.2.2

L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice Hall, 1993. 2.4.1, 2.4.2, 5.2

L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978. 2.2.2, 2.3.1, 2.3.2, 2.3.2, 2.3.3

D. Reynolds and R Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio processing*, 3(1): 72–83, January 1995. 6.11

D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000. 6.11

K. Richmond. *Estimating articulatory parameters from the acoustic speech signal*. PhD thesis, CSTR, University of Edinburgh, 2001. 4.5.2, 5.2

S. Saito and F. Itakura. The theoretical consideration of statistically optimum methods for speech spectral density. Technical Report 3107, Electrical Communication Laboratory, N.T.T., Tokyo, 1966. (in Japanese). 2.2

A. Schmidt-Nielsen and D. P. Brock. Speaker recognizability testing for voice coders. In *ICASSP-96*, volume 2, pages 1149–1152, Atlanta, Georgia, 1996. 3.2

T. Schultz and A. Waibel. The GlobalPhone project: Multilingual LVCSR with JANUS-3. In *Multilingual Information Retrieval Dialogs: 2nd SQEL Workshop*, pages 20–27, Plzen, Czech Republic, 1997. 6.11.2

Y. Shiga. *Precise Estimation of Vocal Tract and Voice Source Characteristics*. PhD thesis, CSTR, University of Edinburgh, 2005. 4.5.2

Y. Shiga and S. King. Estimating detailed spectral envelopes using articulatory clustering. In *5th ISCA Speech Synthesis Workshop*, June 2004. 5.2

H. W. Sorenson. Least-squares estimation: From Gauss to Kalman. *IEEE Spectrum*, 7: 63–68, Jul. 1970. 2.2

S. S. Stevens and J. Volkmann. The relation of pitch of frequency: A revised scale. *Am. J. Psychol.*, 53:329–353, 1940. 2.4.2

Y. Stylianou. *Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification*. PhD thesis, l'Ecole Nationale Supérieure des Télécommunications, 1996. 2.3, 2.3.6, 5

Y. Stylianou. Removing the linear phase mismatches in concatenative speech synthesis. *IEEE Trans. on Speech and Audio Processing*, 9(3):232–239, 2001. 3

Y. Stylianou and O. Cappé. A system voice conversion based on probabilistic classification and a harmonic plus noise model. In *Proc. ICASSP*, pages 281–288, Seattle, Washington, 1998. 3.3.2

Y. Stylianou, O. Cappé, and E. Moulines. Statistical methods for voice quality transformation. In *Proc. EUROSPEECH95*, pages 447–450, Madrid, Spain, 1995a. 3.3, 3.3.2

Y. Stylianou, J. Laroche, and E. Moulines. High-quality speech modification based on a harmonic + noise model. In *Proc. EUROSPEECH95*, Madrid, Spain, 1995b. 3.3.2

Y. Stylianou, O. Cappé, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Transactions on Speech and Audio Processing*, 6(2):131–142, 1998. 3.2

H. Sündermann, D.and Höge, A. Bonafonte, H. Ney, A. Black, and S. Narayanan. Text-independent voice conversion based on unit selection. In *ICASSP*, 2006. 3.1, 6.10

P. Taylor, A. Black, R. Caley, and S. King. Edinburgh Speech Tools. http://festvox.org/festival, 1998. 4.4, 4.5.2, 4.5.4

T. Toda. *High-Quality and Flexible Speech Synthesis with Segment Selection and Voice Conversion*. PhD thesis, Nara Institute of Science and Technology, 2003. 3.3, 3.3.2, 6.4

T. Toda and K. Shikano. NAM-to-speech conversion with gaussian mixture models. In *Interspeech 2005*, pages 1957–1960, 2005. 3.2

T. Toda, A. Black, and K. Tokuda. Mapping from articulatory movements to vocal tract spectrum with gaussian mixture model for articulatory speech synthesis. In *5th ISCA Speech Synthesis Workshop*, June 2004a. (document), 1.2, 4.3.1, 5.2, 5.4.3

T. Toda, A. Black, and K. Tokuda. Acoustic-to-articulatory inversion mapping with gaussian mixture model. In *Proc. ICSLP2004*, pages 1129–1132, Oct. 2004b. 5.2, 5.4.3

T. Toda, A. W Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235, 2007. 3.3, 3.3.2, 3.3.2

T. Toda, A. Black, and K. Tokuda. Statistical mapping between articulatory movements and acoustic spectrum with a gaussian mixture model. *Speech Communication*, 50(3): 215–227, Mar. 2008. 5.2

K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *ICASSP*, volume 3, pages 1315–1318, 2000. 2.4.3

A. Toth. Cross-speaker articulatory position data for phonetic feature prediction. In *Interspeech2005*, Lisboa, Portugal, 2005. 1.2, 1.6, 4.1, 1

A. Toth and A. Black. Visual evaluation of voice transformation based on knowledge of speaker. In *ICASSP*, 2006. 3.5.2, 4.6, 1, 6.10

A. Toth and A. Black. Using articulatory position in voice transformation. In *ISCA SSW6*, 2007. 1.6, 1

E. Uraga and T. Hain. Automatic speech recognition experiments with articulatory data. In *Interspeech 2006*, 2006. 4.1, 4.5.2

M. Wester, J. Frankel, and S. King. Asynchronous articulatory feature recognition using dynamic bayesian networks. In *Proc. IEICI Beyond HMM Workshop*, Kyoto, Dec. 2004. 5.2

A. Wrench. The MOCHA-TIMIT articulatory database. Queen Margaret University College, http://www.cstr.ed.ac.uk/artic/mocha.html, 1999. 1.3, 4.2

A. Wrench. A new resource for production modelling in speech technology. In *Proc. Workshop on Innovations in Speech Processing*, Stratford-on-Avon, 2001. 5.3.2

A. Wrench and K. Richmond. Continuous speech recognition using articulatory data. In *Proc. ICSLP2000*, Beijing, China, 2000. 4.1

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda. The HMM-based speech synthesis system (HTS) version 2.0. In *ISCA SSW6*, 2007. 1.3, 6.14.1

J. Zhang, A. Toth, K. Collins-Thompson, and A. Black. Prominence prediction for super-sentential prosodic modeling based on a new database. In *5th ISCA Speech Synthesis Workshop*, 2004. 5.4.1