TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Victor Popa

Techniques for Spectral Voice Conversion

Julkaisu 1049 • Publication 1049

Tampere 2012

Tampereen teknillinen yliopisto. Julkaisu 1049
Tampere University of Technology. Publication 1049

Victor Popa

Techniques for Spectral Voice Conversion

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB104, at Tampere University of Technology, on the 1st of June 2012, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2012

ISBN 978-952-15-2835-4 (printed)
ISBN 978-952-15-2878-1 (PDF)
ISSN 1459-2045

Abstract

Voice conversion is a speech technology encompassing transformations applied to the speech signal with the purpose of changing the perceived speaker identity from a source voice to a desired target voice. The principal use of voice conversion is to enable synthesis systems to generate speech with customized voices without need for exhaustive recordings and processing. Voice conversion can be realized as a stand-alone task or alternatively, using adaptation techniques with HMM-based synthesis. Although there are many speaker-dependent voice characteristics, voice conversion deals mainly with those acoustic in nature such as the spectral characteristics and the fundamental frequency. In spite of the remarkable results achieved by the state of the art techniques, further challenges remain to be solved in all sub-areas of voice conversion in order to provide excellent quality and highly successful identity conversion. The objective of this thesis is to develop a stand-alone voice conversion system for coded speech and to propose solutions leading to better quality, versatility or efficiency compared to current techniques. The thesis is focused on the conversion of spectral envelopes but other sub-areas of voice conversion such as speech parameterization or alignment are also treated.

The analysis-modification-synthesis system adopted in this thesis is based on a parametric speech model used in a real speech codec but also for the internal speech representation of a concatenative TTS system. This allows an easy integration of voice conversion with communications related and embedded applications developed for coded speech. The validity of this parameterization has been confirmed by using it in an actual voice conversion scheme. In addition, a voicing level control scheme, a speech enhancement technique and a method for automatic speech data collection have been proposed, all of them taking advantage of this parametric framework.

The versatility of voice conversion systems depends on the characteristics imposed on the training data in order to properly estimate a conversion function. Depending on the characteristics of the training data different alignment strategies can be adopted. Although dynamic time warping (DTW) has become almost a standard for the alignment of parallel training data, a new soft alignment technique is proposed for the same purpose. This concept allows probabilistic one-to-many frame mapping and is proven valid in an experiment with an artificial example. For practical reasons the non-parallel case has been attracting increased interest lately. In this thesis, two techniques for text independent alignment are proposed. The first one is based on phonetic segmentation and temporal decomposition and was successfully used in a voice conversion application. The second method uses a TTS to break the non-parallel problem into two parallel conversions which concatenated realize the desired voice conversion.

GMM-based techniques have been the most popular approach for the conversion of spectral envelopes in spite of their problems related to over-smoothing, over-fitting and the lack of temporal modeling. In order to address these issues and improve the GMM-based voice conversion, this thesis introduces several techniques. First, a new measure of the conversion accuracy is proposed, which can be easily computed from the GMM parameters, and is found to be in line with perceptual and objective metrics. The next technique combines a clustering and mode selection scheme with cluster-

wise GMM modeling and is based on the observation that the mapping accuracy improves with the reduction of target data variance. The proposed technique is shown to outperform a comparable approach that uses voicing based clustering. A third technique can be used to adapt an existing well-trained conversion model to a new target speaker using a small amount of data. In a practical evaluation the adapted model outperformed an equivalent model trained exclusively on the reduced data. The continuity issue is addressed in a final idea proposed for future research.

In general the spectral conversion techniques proposed in the literature have been subject to a tradeoff between speech quality and identity conversion and the challenge remains to provide optimal results for both criteria simultaneously. A new spectral conversion technique introduced in this thesis uses bilinear models to decompose the line spectral frequencies (LSF) representation of the spectral envelope into two factors corresponding to speaker identity and phonetic content respectively. In an extensive evaluation on different types and for different sizes of training data this approach was found to perform similarly to a GMM-based conversion even though, in contrast to the GMM, it does not require any tuning. The concepts of contextual and local modeling are also introduced arguing that a more accurate mapping can be achieved if multiple models are fit on relatively small subsets of the full training data rather than using one global model. The validity of the concepts has been verified in practical experiments. Furthermore, a local linear transformation technique is shown to effectively reduce the over-smoothing relatively to a globally trained GMM-based conversion function.

For spectral conversion, several of the proposed techniques can be considered extensions of a baseline vector quantization framework, opening the way for further developments in this direction. In addition to the local linear transformation method which can be easily integrated in this framework, a memory efficient conversion scheme based on multi-stage vector quantization (MSVQ) was also proposed. The experiments indicated substantial memory savings and even accuracy improvement compared to some conventional codebook conversions. A dynamic programming approach to optimizing the frame to frame continuity in a vector quantization framework is also proposed as a future direction.

A final contribution is brought to the existing hybrid technique that combines GMM and frequency warping. The approach is adapted to work with the proposed speech representation and a procedure for automatic formant alignment and warping function calculation is presented.

Acknowledgment

The research presented in this thesis has been conducted during the period 2005-2011 at Nokia Research Center (NRC) and at the Department of Signal Processing of Tampere University of Technology (TUT).

I would like to thank my thesis supervisor, Professor Moncef Gabbouj, for his guidance and kind support throughout the years of this research. I thank the pre-examiners, Prof. Thierry Dutoit (Faculte Polytechnique de Mons, Belgium) and Prof. Yoshikazu Miyanaga (Hokkaido University, Japan), for the time and effort dedicated to read this thesis and for their valuable comments and recommendations in view of the thesis improvement.

Next, I would like to express my gratitude to a number of collaborators and people who have contributed directly or indirectly to this thesis. I am particularly thankful to MSc. Jani Nurminen for his continuous collaboration and support and for our countless discussions. I thank Dr. Jilei Tian and my other co-authors MSc. Hanna Silén, MSc. Elina Helander for the fruitful collaboration and for their valuable contribution to this work. I would also like to thank Prof. Ioan Tabus for his kind help with some punctual discussions.

The funding provided by Tekes (the Finnish Funding Agency for Technology and Innovation) during the work at NRC, by Tampere Graduate School in Information Science and Engineering (TISE) and by the Academy of Finland through the Finnish Programme for Centres of Excellence in Research 2006-2011 (application number 129657) is gratefully acknowledged.

My colleagues and staff members from both NRC and TUT created a good atmosphere and work environment for which I am grateful and I would like to thank them all. Special thanks are due to Virve Larmila, Johanna Pernu, Elina Orava and Ulla Siltaloppi for their kind assistance and help with the practical matters. Furthermore, I am thankful to Tapio Manninen, Francesco Cricri and Marek Miettinen for the pleasant office atmosphere and the good company during lunches and outside the work hours. I also thank Vinod Kumar Malamal Vadakital, Alex Onose and Tapio Manninen for some interesting work related discussions.

I wish to thank all my friends for being there when I need them and for all the good moments and cheerful mood they have brought to me during this long period: Corina, Vinod, Luis, Profe, Jaacan, Panda, Laureano, Fernando, Principito, Marian, Vlad, Mihai, Marius, Umur and many others.

My deepest gratitude goes to my parents Elena and Ionel Popa and to my brother Adrian for their sacrifices and their unconditional love and support which have been an endless source of inspiration and motivation for me to advance towards higher goals and to accomplish this thesis. At the same time I want to thank my girlfriend Miia for being by my side and supporting me towards this achievement with her love and understanding.

Tampere, March 2012 Victor Popa

Contents

List of Figures

List of Tables

List of Abbreviations

ABS	Analysis by synthesis
ANN	Artificial neural network
AR	Autoregressive (model)
BL	Bilinear (model)
CC	Cepstral coefficients
DCT	Discrete cosine transform
DTW	Dynamic time warping
EM	Expectation maximization
EV-GMM	Eigen voice Gaussian mixture model
FD-PSOLA	Frequency-domain pitch-synchronous overlap-add
FFT	Fast Fourier transform
FW	Frequency warping
GMM	Gaussian mixture model
HMM	Hidden Markov model
HNM	Harmonic plus noise model
HSM	Harmonic plus stochastic model
HQ-TTS	High quality text-to-speech (system)
IAIF	Iterative adaptive inverse filtering
LLT	Local linear transformation
LP	Linear prediction
LPC	Linear predictive coding
LP-PSOLA	Linear predictive pitch-synchronous overlap-add
LSF	Line spectral frequencies
MAP	Maximum a posteriori
MBE	Multi-band excitation
MCC	Mel cepstral coefficients
MFCC	Mel frequency cepstral coefficients
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
MLST	Maximum likelihood stochastic transformation
MOS	Mean opinion score
MRTD	Modified restricted temporal decomposition
MSE	Mean squared error
MSVQ	Multi-stage vector quantization
OLA	Overlap-add
pdf	Probability distribution function

PLS	Partial least squares (regression)
PSOLA	Pitch-synchronous overlap-add
RBF	Radial basis function
SD	Spectral distortion
SNR	Signal to noise ratio
STASC	Speaker transformation algorithm using segmental codebooks
STFT	Short-time Fourier transform
STRAIGHT	Speech transformation and representation using adaptive interpolation of weighted spectrum
SVD	Singular value decomposition
SVR	Support vector regression
TD	Temporal decomposition
TTS	Text-to-speech (system)
VAD	Voice activity detection
VLBR	Variable bit rate
VQ	Vector quantization
VTLN	Vocal tract length normalization

Chapter 1

Introduction

In the context of booming modern technology, the presence of computers in our lives became a fact and brought the need for a more natural communication between humans and machines. Speech is the most accurate and natural communication instrument between humans enabling them not only to exchange ideas but also to transmit emotions. As such, speech has been receiving a huge interest from the scientific community and many research areas have emerged to study its structure, production and perception in an effort to design speech based human computer interfaces.

Voice conversion is a relatively new topic in speech research aiming essentially to change the perceived voice in speech signals. This chapter defines the problem of voice conversion, presents the motivation and applicability of such a technology and sets the scope and objectives of this thesis. The chapter continues with a description of the author's contribution and concludes with a brief outline of the thesis.

1.1 BACKGROUND

The two major technologies involved in the communication between humans and computers are speech recognition and speech synthesis. Speech recognition is needed to recognize word sequences in a speech signal independent of the speaker's voice. Speech synthesis, on the other hand, can be defined as the artificial production of speech. The subject of this thesis, voice conversion, is seen as belonging to the speech synthesis framework.

Voice conversion studies modifications of speech signals that transform the original voice into a desired target speaker's voice, changing in this way the perceived speaker but leaving unaltered the uttered content. A voice conversion system performs two main tasks:

1) *Training*: The system determines an optimal transformation between the original (source) and the target voice characteristics.
2) *Conversion*: The transformation is used to convert new source speech.

1.1.1 Speech Production and Speaker Identity

Speech is the result of pressure variations applied by the articulatory system to an air-flow produced by the respiratory system. The air is pushed from the lungs into the trachea and through the vocal folds further into the vocal tract (pharynx, nasal and oral cavities). An aperture of the vocal folds called glottis differentiates between voiced and unvoiced sounds depending on whether it blocks or not the air passage. When the glottis is closed, the air flows between vibrating vocal folds generating voiced sounds. A free passage of the airflow through an open glottis produces unvoiced sounds instead. Specific phonemes are produced by modifying the shape of the vocal tract through articulators such as lips, tongue, jaws and teeth.

An illustrative model of speech production from a signal processing perspective is the source-filter model [1]. The glottal airflow is represented as a source or excitation signal which takes the form of a pulse train for the voiced sounds and the form of a noise signal for the unvoiced. A voiced excitation is characterized by a fundamental frequency or pitch which is determined by the oscillation frequency of the vocal folds. The vocal tract is seen as a resonator cavity that shapes the source signal in frequency, and can be understood as a filter with a specific frequency response. Its resonator frequencies are called formants. The speech signal is the result of filtering the glottal source signal through the vocal tract filter. The specific formant structure of each phoneme is obtained gradually by changing the position of the articulators.

Speech carries multiple types of information. It encodes linguistic information in the form of a phonetic sequence based primarily on the characteristics of the vocal tract (formants) and glottal source (voiced/unvoiced). Moreover, the pitch contour differentiates between affirmative, negative and interrogative utterances, the stress marks individual words by a local peak in the pitch contour and prosodic features such as intonation, speaking rate or rhythm are capable to convey information about the emotional state of the speaker. Very importantly, the speech also carries information about the speaker identity.

For the purpose of voice conversion it is essential to understand what factors determine the speaker identity. They can be *linguistic* or *acoustic*.

1) The dialect, together with the speaker's preference for particular syntactic and lexical patterns are examples of *linguistic* factors relevant for the individuality. These factors are situated at the message level and are generally influenced by the social class, region of birth or residence, age of the speaker.
2) The characteristics that can be measured directly from the speech signal and are independent of the underlying message are called *acoustic* factors. They can be further divided into *prosodic* and *spectral*.
    a) The *prosodic* features relate to the speaking style and include phoneme duration, pitch contour (intonation), energy (stress) [2], and voice quality. Voice quality refers to the characteristics of the voicing sound source ranging from laryngealized to normal and breathy phonation and conveying information about the speaker's emotions, mood and attitude [3][4].
    b) At *spectral* level the speech sound is described in terms of formant locations and bandwidths, spectral tilt and excitation of the vocal folds [5].

The difficult task of modeling personal linguistic properties is currently omitted in the existing voice conversion systems which mainly deal with the acoustic level. Most of the actual systems are in fact focused exclusively on the spectral level.

1.1.2 Speech Models and Modifications

The acoustic modification of speech can be achieved in three basic ways by manipulating independently its speed, fundamental frequency or formants. The first two categories correspond to the prosodic level while the third corresponds to the spectral level.

- *Time-scale modification* changes the speech duration while preserving its fundamental frequency and spectral properties.
- *Pitch modification* changes the fundamental frequency of a speech signal while preserving the original duration and spectral properties.
- *Spectral modification* changes the formant structure while preserving the fundamental frequency and original duration of the signal.

In order to perform voice conversion, analysis/synthesis methods are necessary to give speech a parametric representation and to be able to synthesize speech from a modified parametric representation. Many analysis/synthesis methods have been proposed in the literature, some in the time domain [6][7][8], some in the frequency domain [9][10][11], and others in the time-frequency domain [12][13][14]. Next, we briefly present some of the most important and relevant approaches.

Pitch-synchronous overlap-add (PSOLA) [6][7][8] uses overlapping windows and operates directly on the waveform on a frame basis permitting flexible manipulations of duration, pitch and formants. To change the time-scale, for instance, frames are either repeated or dropped leaving the pitch marks unchanged. A modification of pitch is achieved by adjusting the spacing between the pitch marks instead. For spectral envelope manipulation the variants FD-PSOLA and LP-PSOLA [15] can be used. In spite of a clear sound the method is less suitable for fine modifications [16] compared to other techniques [17] and it may introduce artifacts e.g. in the synthesis of modified unvoiced sounds [18] [19].

Other methods for speech modification are based on the source-filter model described previously and use different ways to estimate the excitation and the parameters of the vocal tract filter. Assuming an all-pole model, the formant structure can be manipulated by changing the formant locations and magnitudes [20][21]. Alternatively, frequency warping of spectral envelopes [22] has been proposed. These techniques have obvious limitations. The pole modification, for instance, cannot control formant bandwidth and amplitude independently.

Another analysis/synthesis technique was developed on a sinusoidal speech model by McAulay and Quatieri [23][24]. The speech is represented as a sum of time-varying sinusoids whose amplitude, frequency and phase parameters are estimated from the short-time Fourier transform using a peak-picking algorithm. This framework lends speech to time and pitch-scale modification producing high-quality results.

1.1.3 Spectral Modifications

The modification of spectral properties directly affects perception and represents one of the fundamental problems of voice conversion. The spectral processing is applied to spectral envelopes obtained from a time-frequency speech representation in order to change their formant structure. The challenge, here, concerns the analysis/synthesis technique which should ensure high quality reconstruction of speech from modified spectral features. Many methods have been proposed but the largest impact was made by three most popular techniques based on Gaussian mixture models (GMM) [25], mapping codebooks [26] and frequency warping [27]. Most of the existing methods are

based on statistical models trained on large databases. Other methods try to control the shape of the spectral envelope by changing the pole locations and amplitudes.

1.1.4 Types of Voice Conversion

a. Stand-alone voice conversion

The most common form of voice conversion is a stand-alone system with the general structure presented in Chapter 2 (Figure 2.1). In the training phase, a speech corpus recorded from a source and a target speakers is used to train a conversion function which will be used for the transformation. First, the speech signals are analyzed using a proper speech model that allows flexible signal manipulation. Then each analyzed frame is parameterized with a set of features suitable for a good conversion of the acoustic characteristics. The correspondence between these acoustic characteristics of the two speakers is determined from the training data during the alignment process. Finally, a conversion function is estimated for the spectral and/or prosodic features. During the conversion phase, the system analyzes and parameterizes the new utterances of the source speaker with the same scheme used in training and then transforms them by applying the trained function.

b. Voice conversion in HMM-based speech synthesis

Voice conversion is typically formulated as a standalone problem with the typical implementation presented earlier in this section. However, the concept of voice conversion and the methods to realize it may significantly change in some particular situations. This is the case with the HMM-based speech synthesis systems [28] which use HMMs to generate an optimal parameter sequence from which speech can be synthesized. Voice modification is achieved using adaptation techniques like MLLR (maximum-likelihood linear regression) which adapt the HMM of the source speaker to maximize the likelihood of the target speaker's data. The approach to voice conversion is conceptually different in this context. Despite promising results [29] HMM synthesis has certain qualitative limitations due to the statistical parameter generation. The best quality is currently achieved by concatenative text to speech (TTS) systems which produce speech by concatenating pre-recorded speech units.

1.1.5 Evaluation of Voice Conversion Systems

The success of voice conversion is measured with respect to two aspects:

- Identity conversion i.e. similarity of the converted voice and the target voice
- Sound quality which describes the level of distortions and artifacts.

The methods of evaluation may be: *subjective* and *objective*. Objective evaluations use mathematically defined metrics such as mean squared error (MSE), spectral distortion and signal to noise ratio (SNR) to assess the converted speech objectively. This kind of evaluation is repeatable i.e. produces always the same results for the same data and computationally inexpensive. On the other hand, objective measurements do not relate well to human perception. Subjective evaluations are more relevant from a perceptual point of view being based on listening tests. On the downside they lack stability and repeatability depending strongly on factors such as listener's mood and familiarity with the field or on other samples evaluated at the same time.

1.2 APPLICATIONS

Voice conversion technology opens the way to a large number of applications most of which are closely related to speech synthesis. The applications range from personalized text to speech voices to entertainment and security related applications [30][18].

The first important application of voice conversion is as an extension module to a text-to-speech system (TTS) transforming the pre-recorded voice to a desired target voice. This way personalized and branded TTS voices may be created inexpensively [31].

Potential applications in the entertainment industry include movie dubbing, generating speech with voices that no longer exist, disguising speaker identity, creating virtual voices for characters in a videogame or creating compact “audio books” in text format where the narrator and characters involved in dialogues each use their individual voice. Furthermore e-mails and SMS messages could be “read” to us with the sender’s voice.

In speech-to-speech translation voice conversion can recover and use the original speaking voice to synthesize the translated utterance helping in this way the listeners to easily identify the speaking person [32]. Voice conversion will be used as an extension module to the existing speech recognition, machine translation and speech synthesis components.

An educational application is related to learning new languages by listening to one’s own voice speaking the new language with proper pronunciation and intonation. Apart from that, a simple rate reduction of a native speech is a useful tool.

From a medical perspective the voice conversion provides a means to enhance the speech quality for persons with speaking disabilities or hearing disabilities [33][34].

Applications of voice conversion can be found also in other speech technologies. In speech coding time-scale modifications can be used for data and bandwidth reduction. In speech recognition speaker normalization e.g. by vocal tract length normalization improves the accuracy rate. In speech enhancement conversion techniques could be used to improve intelligibility and speech quality.

Finally, Eide and Picheny [35] used voice conversion for the normalization of speech databases in order to increase the amount of data available as needed for the construction of a concatenative speech synthesis system.

1.3 SCOPE AND OBJECTIVES OF THE THESIS

This thesis treats voice conversion as a standalone problem and does not involve techniques associated with HMM-based synthesis.

Like most studies on voice conversion this work addresses the modification of the acoustic characteristics of speech, namely prosodic and spectral attributes, and excludes the linguistic factors which are very difficult to model. The thesis effectively focuses on the spectral modification which is considered the core problem of voice conversion and only slightly covers the prosodic aspects.

The challenge of spectral conversion is to modify the spectral structure without compromising the speech quality. The problems with the existing spectral modification techniques are related primarily to continuity. These methods are based on frame by frame processing which ignores the relationship between neighboring frames and they fail to model the temporal evolution of the parameters. The second problem with these methods is the ineffective spectral transformation. The pole modification methods, for instance, do not provide full control of the spectral envelope being impossible to control

a formant bandwidth and amplitude independently. The frequency warping methods have problems preserving the shape of modified spectral peaks, controlling the bandwidths of close formants, allowing formants to merge or even controlling the formant amplitudes. Furthermore GMM-based methods suffer from an over-smoothing of the converted spectrum while codebook methods suffer from discontinuity.

This work was carried out based entirely on a parametric representation used by a variable bit-rate (VLBR) speech codec in an attempt to combine voice conversion with the advantages of an efficient compression. This parameterization is based on an underlying source-filter speech model and uses linear prediction (LP) to model the vocal tract contribution and sinusoidal modeling to represent the excitation.

An important part of this research was accomplished using parallel training data due to the existence of well-studied algorithms for alignment but the work also proposes an alignment technique for non-parallel data.

Starting from these premises the objectives of this thesis can be stated as follows:

- To develop speech modification techniques over the VLBR framework facilitating the integration of voice conversion with a TTS system using a similar speech representation. This system would combine voice conversion with the attractive compression properties of speech coding.
- To improve the state-of-the-art by addressing the above mentioned problems and to propose new spectral conversion methods capable to correctly map the speaker identity and produce a high quality speech.
- To study and improve the sub-processes of a voice conversion system including alignment, model estimation and the evaluation.
- To study and propose conversion methods trainable from reduced data.
- To develop suitable algorithms for all possible conversion scenarios: text dependent and text independent (intra-lingual or cross-lingual).
- To develop a framework capable to represent separately the speaker dependent and content dependent information.
- To investigate fast and low footprint solutions to voice conversion.

Summarizing, the goal is to develop voice conversion algorithms over a VLBR speech codec representation in order to combine it with attractive compression properties. The thesis is focused on improving the voice conversion in terms of both speech quality and identity mapping.

1.4 AUTHOR'S CONTRIBUTIONS

The research conducted to address the stated objectives led to findings and solutions to a variety of tasks involved in the process of voice conversion.

Some of the contributions are directly related to the VLBR speech codec framework:

- A new framework for voice conversion based on a speech codec representation which adds the advantage of efficient compression.
- A method to correct voicing problems particular to the proposed speech codec framework induced by spectral conversion.
- A practical solution to collect speech data for voice conversion over phone calls facilitated by the proposed speech codec framework.

- A method to improve the quality of the models trained and to reduce the distortions perceived during the non-speech parts of the converted samples, which takes advantage of the proposed speech codec framework.

Several ideas have been proposed to upgrade the performance of codebook based methods by reducing the memory footprint and improving the spectral smoothing.

- The idea to use multi-stage vector quantization (MSVQ) for voice conversion in order to reduce the memory footprint of the conversion model. This may additionally improve the accuracy and reduce the complexity.
- An extension of the vector quantization framework with a local linear transformation technique was shown to reduce the over-smoothing and obtain better perceptual results than the popular GMM-based approach.
- A dynamic programming approach to model temporal evolution and use temporal information to improve the spectral continuity.

Other novelties of this thesis are related to the various training scenarios.

- A soft alignment method that allows multiple mappings per frame and assigns alignment probabilities to source-target frame pairs.
- An alignment scheme for text-independent data (both intra- and cross-lingual). The method is based on phonetic segmentation and on speech coding techniques for temporal decomposition [16].
- A practical solution to text independent voice conversion by using a TTS system to generate in turn speech parallel to the source utterances and then speech parallel to the target utterances. The source-to-TTS and TTS-to-target conversion functions are computed using methods for parallel data and then concatenated.

The next contributions resulted from the study of the GMM-based conversion.

- An efficient scheme to re-estimate a conversion function by using limited amount of data from a new speaker to adapt an existing well trained function.
- A method to objectively evaluate the performance achievable by a GMM directly from its parameters without performing an actual conversion. A direct relationship was found between a proposed trace measure and the mean squared error scores.
- A clustering scheme for the training data using auxiliary features besides the spectral ones to achieve minimal intra-cluster variability. Each cluster will train a different model. A classifier is used to select the correct conversion model.
- The idea to extract and convert dynamic features and use them to improve the conversion of the static features using optimization techniques.

A major contribution is to propose a new thinking about voice conversion:

- A new formulation of the spectral envelope as product of a style factor representing the voice characteristics and a content factor representing the underlying phonetic content. The method showed robust performance with reduced training sets. This separation may be useful also for speaker identification and speech recognition tasks.

Finally, other contributions are:

- To propose and test the concepts of contextual and local modeling, a scheme in which multiple models are trained on possibly overlapping subsets of the training data.
- A hybrid GMM-frequency warping voice conversion system based on parametric speech representation and a technique for automatic calculation of the warping function.

1.5 THESIS OUTLINE

This section presents the organization of the thesis summarizing the chapter contents.

Chapter 2 introduces the fundamentals of voice conversion starting with the mechanisms of human speech production and continuing with the functional parts of a voice conversion system: the analysis/synthesis method, the speech parameterization, the alignment and the model estimation. The state of the art solutions are critically discussed emphasizing their limitations.

In Chapter 3 a parametric framework for voice conversion based on a VLBR speech codec is presented. This chapter gives details about the speech representation and parameter estimation and demonstrates the framework's operability in a GMM based conversion. A practical use of the favorable compression properties in data collection is presented together with a noise attenuation technique and a voicing level correction scheme. The publications relevant for this chapter are [36], [37] and patent applications [38] and [39].

Chapter 4 addresses the alignment problem proposing solutions to all training scenarios. For text dependent data, it proposes and demonstrates the concept of soft alignment. For text independent data two other schemes are introduced: one based on phonetic segmentation and temporal decomposition and another based on TTS and cascading of a source-to-TTS conversion model with a TTS-to-target model. This chapter is based on publication [40] and patent applications [41] and [42].

Chapter 5 and Chapter 6 are dedicated to the central problem of voice conversion and of this thesis, the spectral conversion. Chapter 5 presents a group of algorithms aimed to improve different aspects of the GMM based voice conversion: a method for efficient evaluation of GMM-based transformations, a conversion scheme based on clustering and mode selection, a technique for efficient re-estimation of conversion models from limited data and a conversion approach based on temporal dynamic features. These ideas have been published in [43][44][45][46].

Among the techniques proposed in Chapter 6, some can be regarded as extensions of a vector quantization framework. These include a memory saving scheme, a conversion approach based on local linear transformations and a dynamic programming scheme that optimizes the temporal evolution of spectral parameters. In addition, the chapter formulates a new perspective on voice conversion showing how factor analysis tools, and in particular bilinear models, can be used to decompose the speech signal into two factors representing the voice characteristics and phonetic content respectively. The concept of contextual modeling is also briefly introduced. The last part of the chapter describes how the hybrid GMM-frequency warping conversion scheme should be adapted to work with a parametric speech representation and proposes an automatic technique for formant alignment and warping function calculation. The chapter is based on publications [47][40][48] and patent applications [49].

Chapter 7 summarizes the contributions of this thesis, presents the remaining open issues and indicates directions for future research.

Chapter 2

Stand-Alone Voice Conversion System

The voice conversion scenario in which a mapping function is learned from some initial training data of the source and target speakers is referred to as stand-alone voice conversion. This scenario has been the main focus of the voice conversion research until now and will be presented in more detail in this chapter together with the most important results proposed in the literature.

A typical stand-alone voice conversion system is presented in Figure 2.1 and consists of two modules. The training module finds an optimal speech transformation based on the training data from the source and target speakers while the conversion module applies this transformation to convert new utterances of the source speaker.

A very important role in the design of a voice conversion system is played by the speech model used to analyze the input signal and to re-synthesize the modified speech. A good analysis/synthesis framework has to permit flexible spectral modifications without compromising the quality of the reconstructed speech signal. An overview of the most common frameworks is given in section 2.1.

Voice conversion systems typically process speech on a frame basis. Because it is difficult to convert the speech in the form resulted from the analysis (signal windows, short-term spectra etc.), all voice conversion systems parameterize speech frames in order to conveniently represent the identity information and to simplify the training and conversion. The representation of speech in a parametric domain is called *feature extraction*. The extracted features are often required to have good interpolation properties. The typical features used in voice conversion are presented in section 2.2.

In order to learn a mapping function from the training data it is necessary to determine a correspondence between training speech units of the two speakers. This correspondence is usually established between acoustic classes or between individual frames in a process known as *alignment*. An alignment in the strict sense may also be omitted through model adaptation techniques which would, for instance, adapt an already trained transformation. The alignment type largely depends on the properties of the training data and on the particular spectral transformation used by the voice conversion system. The classification of training corpora and alignment methods are presented in section 2.3.

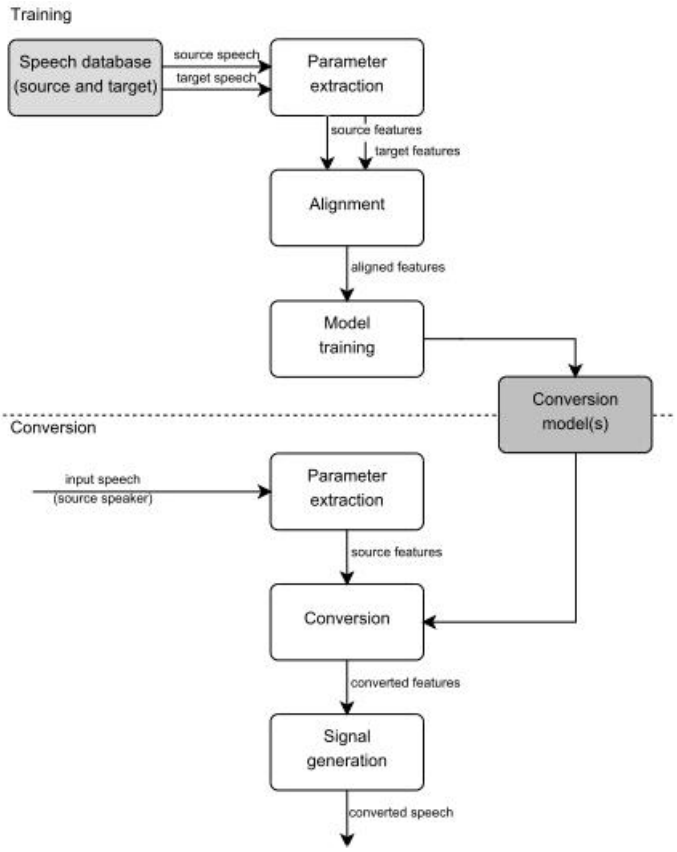

Figure 2.1: Block diagram illustrating a stand-alone voice conversion system. The training phase generates conversion models based on training data, which includes speech from both source and target speakers. In the conversion phase, the trained models can be used for converting unseen utterances of the source speech. (from [50])

After feature extraction and alignment, the *model estimation* block calculates the transformation function which optimally converts the speaker dependent characteristics of the speech signal. The transformations may occur at two levels: prosodic and spectral. In terms of prosodic conversion, a simple range scaling and mean shifting of the pitch level is often considered sufficient for identity mapping. More complex alternatives for converting pitch contours, durations and energy have also been proposed in the literature. The spectral conversion is the most important part of a voice conversion system. Section 2.4 introduces the most typical spectral transformations such as mapping codebooks, frequency-warping functions, neural networks or probabilistic linear transformations.

As shown in Figure 2.1 the transformation functions resulted from the training process are used to convert new input utterances from the source speaker. Given the transformations, the *conversion* process is independent of the training and can be implemented as a separate module in a voice conversion system. Frame by frame, the signal goes through feature extraction, parametric conversion and re-synthesis. The actual conversion is performed in the parametric domain.

2.1 ANALYSIS / SYNTHESIS FRAMEWORKS

An important part in the design of a voice conversion system is represented by the speech model used for the analysis of input signals and reconstruction of the modified signals. A speech model suitable for voice conversion should have the following characteristics [18]:

- It provides high fidelity reconstruction of the signal from the model parameters (copy synthesis).
- It can operate modifications of the prosodic attributes of speech such as pitch, duration and intensity without introducing artifacts.
- It supports flexible spectral modifications without degrading the quality of the synthesized speech.

The first two characteristics guarantee that the speech model is good for synthesis purposes. There is a close relationship between the voice conversion algorithms and the underlying synthesis system and their correct interaction is important when transforming the acoustic features. In fact, the analysis-synthesis process may introduce artifacts in the converted signal. The most common speech models used for synthesis and voice conversion are presented next.

2.1.1 PSOLA Methods

TD-PSOLA [7] is a very popular synthesis technique which allows artifact-free prosodic modifications and provides high-quality synthetic speech. It operates by sampling windowed portions of the original signal and then re-synthesizing them with a basic overlap-add procedure. However, as the speech modification is operated directly from the samples, the method lacks control over spectral envelopes which makes it unsuitable for voice conversion.

For spectral manipulations, some other implementations of PSOLA are preferred. Unlike TD-PSOLA, FD-PSOLA technique [15] modifies the speech signal in the frequency domain, therefore allowing for easy spectral manipulation. In LP-PSOLA [15], the PSOLA technique is combined with a residual-excited LPC model of speech. The speech signal is separated into a time-domain excitation and a time-varying spectral envelope. The modifications are operated on the excitation signal and the result is combined with re-synchronized spectral envelopes in order to generate the transformed speech. These variants of PSOLA are more suitable for voice conversion and have been used in some of the systems proposed in the literature [22][51][52][53][54][55].

2.1.2 Sinusoidal Models

In a sinusoidal model the speech waveform is represented locally as a sum of sinusoids whose parameters vary with time. A harmonic model represents a special type of sinusoidal model whose sinusoids are estimated only at frequencies which are multiples of the local fundamental frequency. The sinusoidal model is suitable for all kinds of voice transformation for many reasons:

- It provides high quality for speech reconstruction and prosodic modification.
- The model parameters contain information about both the waveform and the spectrum. They can be used to derive estimates of the magnitude and phase spectral envelopes and therefore allow flexible spectral manipulation and voice conversion.

- The model characteristics are suitable for concatenative speech synthesis because it allows the suppression of the waveform and smoothing of the spectral discontinuities at the transition boundary between two adjacent units.
- It supports data compression for embedded systems.
- It is compatible with most of the voice conversion methods discussed later in this chapter and as such it has been adopted by many voice conversion systems [25][31][56][57][58].

McAulay and Quatieri

Some of the most relevant contributions to the sinusoidal modeling of speech were made by McAulay and Quatieri. Originally the speech was modeled as a sum of time-varying sinusoids whose amplitudes, frequencies and phases were measured at a constant frame rate using a simple peak-picking algorithm over the STFT [23].

$$s(t) = \sum_{l=1}^{L(t)} A_l(t) cos\varphi_l(t) \tag{2.1}$$

In equation (2.1) $A_l(t)$ and $\varphi_l(t)$ represent the instantaneous amplitude and phase of the l^{th} sinusoid, respectively, and $L(t)$ is the number of sinusoids.

In a later work by the same authors [24][59] the speech signal $s(t)$ is assumed to be the result of passing a glottal excitation signal $e(t)$ through a linear time-varying filter that models the characteristics of the vocal tract, the excitation being represented as a sum of sine waves.

ABS/OLA, George and Smith

The Analysis-by-Synthesis/Overlap-Add (ABS/OLA) sinusoidal model developed by George and Smith [60][13][14] is particular in some respects compared to others. First, it uses a constant frame rate and an ABS procedure to determine the parameters of the sinusoids. Considering that $l-1$ frames have been detected and subtracted one by one from the k-th signal frame, the next sinusoid l is found by adjusting its parameters $\{A_l^{(k)}, \omega_l^{(k)}, \varphi_l^{(k)}\}$ to best fit the remaining residual such as to minimize the energy of the estimation error. In practice the best combination of parameters is found by evaluating the error at uniformly spaced candidate frequencies for which the optimal amplitude and phase are calculated using least-squares optimization. Another particularity of this model is that the time-varying waveform is generated by overlapping frames which are sums of constant-amplitude constant-frequency sinusoids

$$s[n] = \sigma[n] \sum_k w[n-kN] s^{(k)}[n-kN], \quad s^{(k)}[n] = \sum_{l=1}^{L^{(k)}} A_l^{(k)} \cos\left(\omega_l^{(k)} n + \varphi_l^{(k)}\right) \tag{2.2}$$

Here, $w[n]$ represents the window used for OLA, $\sigma[n]$ denotes a gain variable in time, and N is the number of samples equivalent to the analysis frame rate; $s[n]$ is the modeled speech waveform, k is a window index, $s^{(k)}[n]$ represents an approximation of the speech signal in the k^{th} frame and n is the current index in the signal; $A_l^{(k)}$, $\omega_l^{(k)}$, $\varphi_l^{(k)}$ and $L^{(k)}$ denote the amplitudes, frequencies, phases and number of sinusoids modeling the k^{th} frame, respectively.

Although the sinusoidal model can be considered in general an attractive speech representation, the large number of sinusoidal parameters makes the spectral manipulation more difficult than for

example in the case of the source-filter model introduced later. Moreover, the sinusoidal model does not provide control over the formant frequencies and bandwidths [61].

Regarding the aperiodic component of speech, in pure sinusoidal systems it appears as unvoiced bands modeled by similar sinusoidal parameters, but special techniques for frequency manipulation or frequency dithering are applied to preserve the noisy nature of these components. The hybrid models introduced next, use a separate stochastic model to describe the aperiodic component.

2.1.3 Hybrid Models

Hybrid models decompose the speech into a deterministic part and a stochastic part. The deterministic part is described with a sinusoidal or harmonic model while the stochastic part models the aperiodic components of the signal which are not well represented by sinusoids. This model has the advantage that the two components which are different in nature can be handled differently. In voice conversion such a differentiated treatment is beneficial considering that the transformation of voiced segments is much more important for the identity conversion than the transformation of unvoiced segments (where the deterministic component does not exist) [62]. Moreover, the voice quality can be manipulated to a certain degree by adjusting the energies of these two components in the voiced segments. On the other hand, this kind of signal decomposition and finding an appropriate transformation function for each component is not straightforward.

Griffin and Lim, multiband excitation vocoder

This model [63] represents the short-time spectrum of speech as the product of an excitation spectrum and a spectral envelope. The spectral envelope is a smooth approximation of the speech spectrum and the excitation spectrum is represented by a fundamental frequency, a voiced/unvoiced (V/U) decision for each harmonic of the fundamental frequency, and the phase of each harmonic considered voiced. The large number of frequency bands represents a difference to previous simpler models which used at most three bands. An analysis by synthesis approach is used to estimate the model parameters while the synthesis is carried out in time domain for the voiced portion of speech and in frequency domain for the unvoiced portion of speech.

A technique based on this model is the multi-band re-synthesis overlap and add [64] (MBROLA) which avoids the pitch marking and allows spectral interpolation between voiced portions of the speech signal.

Harmonic plus noise model

HNM is based on a pitch-synchronous decomposition of the speech signal into a harmonic part and a noise part. The analysis windows are set at a pitch-synchronous rate during the voiced parts of the signal and at a constant rate for the unvoiced regions. For the voiced frames, the spectrum is divided into two bands separated by a time-varying maximum voiced frequency. The low band consists of harmonically related sinusoids with slowly varying amplitudes and frequencies. The high band is determined by filtering a white Gaussian noise with a time-varying all-pole filter and modulating the result with a time-domain energy envelope. The pitch of the signal and the maximum voiced frequency are both estimated in the first step of the analysis using a time domain pitch detector [65]. In the voiced frames, the amplitudes and phases of the harmonic component are determined by minimizing a weighted time-domain least-squares criterion while the noise component

is represented with an AR model. A voiced frame is fully represented by its fundamental frequency, the number of harmonics, the discrete cepstrum coefficients, the phase envelope, the reflection coefficients of the AR filter and the gain of this filter. In contrast, an unvoiced frame is described only by the AR filter and its gain.

2.1.4 STRAIGHT

Based on the source-filter model of speech presented later in this section, STRAIGHT [66] provides flexible speech manipulation by separating the speech information into mutually independent source and filter parameters. STRAIGHT uses F_0-adaptive spectral analysis combined with a surface reconstruction method in the time-frequency region, and an excitation source design based on phase manipulation. It allows very high manipulation factors for pitch and duration, without significant quality degradation.

The procedures are grouped into three subsystems; a source information extractor, a smoothed time-frequency representation extractor, and a synthesis engine consisting of an excitation source and a time varying filter.

F_0 extraction based on instantaneous frequency produces reliable and smooth F_0 trajectories. (Conceptually *instantaneous frequency* may be interpreted as the frequency of a sine wave which locally fits the signal under analysis.) The other source information extracted is the aperiodicity measure. A F_0-adaptive spectral smoothing based on a cardinal B-spline basis function and a complimentary F_0-adaptive time window effectively remove interferences due to signal periodicity from the time-frequency representation of the signal. The time varying filter is implemented as the minimum phase impulse response calculated from the smoothed time-frequency representation through several stages of FFTs. This implementation also enables suppression of “buzz-like” timbre, which is common in conventional pulse excitation, by introducing group delay randomization in the higher frequency region.

2.1.5 Source-Filter Model

The source-filter model introduced by Fant [1] forms the basis for many popular speech production models. According to this model the speech signal can be seen as a source or excitation signal (the glottal source, or noise generated at a constriction in the vocal tract), filtered with the resonances in the cavities of the vocal tract downstream from the glottis or the constriction. Therefore, a speech signal is represented as follows.

$$S(z) = E(z)G(z)V(z)R(z) \tag{2.3}$$

where $S(z)$ is the acoustic speech waveform, $E(z)$ is the excitation, $G(z)$ is the glottal model, $V(z)$ is the vocal tract filter, and $R(z)$ is the lip-radiation impedance. The excitation is an impulse train with the same frequency as the pitch for the voiced sounds and random noise for unvoiced sounds. In some cases, it is convenient to combine the three latter factors as a single transfer function $H(z)$.

By modeling the vocal tract as a cascade of lossless acoustic tubes a linear model for speech production can be derived.

Linear prediction model

Linear prediction represents a powerful speech analysis technique based on the concatenated lossless acoustic tube model. Under this assumption the composite spectral effects of glottal excitation, vocal tract, and lip radiation are represented by a time-varying all-pole filter with the transfer function $H(z)$ of the form.

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{2.4}$$

where $S(z)$ and $U(z)$ are the z-transforms of output and input signals, respectively. G and a_i denote the gain and coefficients of the filter, respectively. If the order p of the LP filter is high enough to capture the spectral envelope of speech, this all-pole model performs a good reconstruction of speech for all speech sounds when it is excited by an accurate enough input signal (excitation). In the simplest synthesis structure, the filter is excited by an impulse train for voiced speech and by random noise for unvoiced speech. In addition to the traditional two-state excitation model some other open-loop as well as analysis-by-synthesis excitation models have been proposed in the literature [67]. Recently, a deterministic plus stochastic model of the residual signal was shown to convey important speaker characteristics and enhance the synthesis quality [68]. The main advantage of the linear prediction model is that the filter coefficients, a_i, and gain parameter, G, can be estimated in a computationally efficient manner using linear predictive analysis.

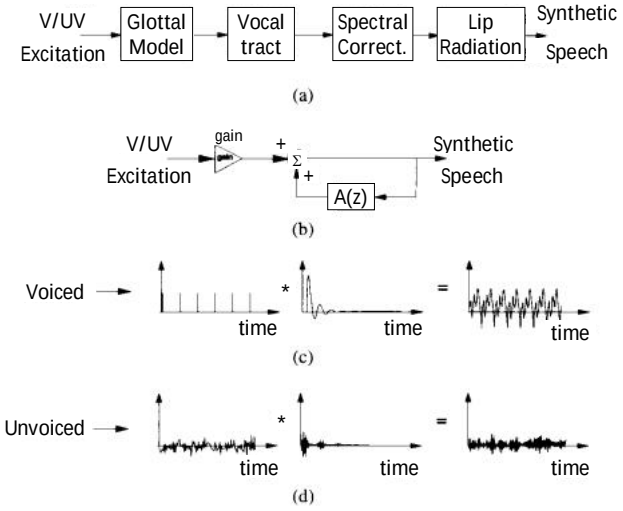

Figure 2.2: Linear speech models and voiced/unvoiced speech representations. (a) Fant's speech production model. (b) All-pole source-system model. (c) Graphical representation of voiced speech production. (d) Graphical representation of unvoiced speech production. (from [67])

In a p-order linear predictor the present sample of the speech sequence is predicted from a linear combination of p past samples, as:

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{2.5}$$

where $\hat{s}(n)$ is the predicted sample and a_i represent the LP coefficients. The prediction parameters are calculated by minimizing the mean squared error e according to:

$$\frac{\partial e}{\partial a_i} = 0, \qquad for\ i = 1,2,\dots,p \tag{2.6}$$

where

$$e = E[(e(n))^2] = E[(s(n) - \hat{s}(n))^2] \quad (2.7)$$

and $E[\cdot]$ is the statistical expectation. The LP coefficients a_i are determined by solving the system of equations (2.6) with either the autocorrelation method [69][70][71] or the covariance method [72].

One of the major issues in LPC is the quantization of the LP parameters [73][74]. Since the direct quantization of the LP coefficients may lead to instability of the synthesis filter some alternative representations have been derived. A widely used representation is the Line Spectrum Frequencies (LSF).

Line spectral frequencies (LSF)

The LSF coefficients are calculated using a symmetric polynomial, $P(z)$, and an anti-symmetric polynomial, $Q(z)$, obtained from the prediction error filter $A(z)$ as follows.

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \quad (2.8)$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}) \quad (2.9)$$

$$\text{where } A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \quad (2.10)$$

The polynomials $P(z)$ and $Q(z)$ have some particular properties [75]:

- All their roots are situated on the unit circle.
- The roots of $P(z)$ and $Q(z)$ are interlaced.
- If the previous conditions hold after quantization or interpolation, then the minimum phase property of $A(z)$ is preserved.

From the first property, we observe that the roots of $P(z)$ and $Q(z)$ can be expressed in terms of the frequencies ω_i (as $e^{j\omega_i}$). These ω_i frequencies are called LSFs and can be derived from equations (2.8) and (2.9) in several ways, for example, by applying a discrete cosine transformation [76] or using Chebyshev polynomials [77].

Being closely related to the speech formants, the LSFs have the advantage that they can be coded using perceptual quantization methods. On the other hand, the computational complexity represents a drawback.

It should be noted that the source-filter model has certain limitations, and cannot describe all types of speech production [16][67]. The lossless assumption for the acoustic tubes used to model the vocal tract is inexact because in reality, the vocal tract is not lossless [16]. Moreover, sounds produced by the nasal tract cannot be explained by an all-pole model. Also, there are certain forms of speech, such as voiced fricatives, which require dual excitation modes. However, this model forms the foundation for later models which have attempted to account for additional factors, for example improved modeling of nasalities was achieved with pole-zero models [78][79][80].

2.1.6 Glottal / Articulatory Models

Glottal inverse filtering [81][82] is a procedure used to estimate the glottal volume velocity waveform, which acts as source of the voiced speech, from speech pressure signals. This procedure is equivalent to obtaining the glottal volume velocity $G(z)$ from the equation

$$G(z) = \frac{S(z)}{V(z)L(z)} \quad (2.11)$$

where $S(z)$, $V(z)$ and $L(z)$ represent z-transforms of the speech waveform, vocal tract and lip radiation, respectively. Considering the lip radiation effect $L(z)$ to be a fixed differentiator [83], only the vocal tract parameters have to be estimated in order to compute the glottal flow from the speech pressure signal with the previous formula.

Note that the source signal determined in equation (2.11) is different from the residual excitation that appears in the linear predictive (LP) model. In the conventional LP analysis a single all-pole filter is estimated which combines the spectral effects of glottal excitation, vocal tract and lip radiation while the excitation is either an impulse train or a spectrally white noise. On the other hand, the excitation given in equation (2.11) is not spectrally white and is allowed to have spectral envelopes of different decays.

In the past decades several methods have been proposed for estimating the glottal flow from speech, see e.g. [84][85][86][87][88][89][90]. One automatic implementation of glottal inverse filtering is known as iterative adaptive inverse filtering (IAIF) [84] [91].

Iterative adaptive inverse filtering (IAIF)

IAIF [84][91] performs automatic decomposition of the voiced speech into vocal tract transfer function and glottal source in two phases. First, it computes a first-order all-pole model to obtain a preliminary estimate for the glottal contribution. In the second phase, it applies a higher order all-pole model which can estimate the contribution of the glottal source more accurately.

2.2 COMMON PARAMETERIZATIONS FOR SPECTRAL CONVERSIONS

It is usually difficult to convert voices directly from the representations resulted from the analysis (signal periods, short-time spectrum samples, LPC coefficients, amplitudes/frequencies/phases). The role of parameterization is to find a set of features that capture speaker identity and can be easily converted while maintaining a high quality of the converted speech. Low dimensionality and good interpolation properties of these features are desirable and beneficial for the purposes of voice conversion. The feature extraction, conversion and re-synthesis is typically operated on a frame-by-frame basis (segmental level) and only in few cases extended to supra-segmental features like pitch contours. The existing studies [92][93][94][95][96] indicate short time spectral envelope and pitch to have the highest contribution to the individuality. Consequently the vast majority of the voice conversion systems are focused on the transformation of these characteristics. This section presents some of the most common types of parameterization used in voice conversion.

Formants: The formant frequencies, bandwidths and intensities are very attractive spectral descriptors but their estimation is usually challenging.

LSF: The LSFs are derived from the LP coefficients and for the purpose of spectral modification they have several attractive properties compared to other spectral representations which made them a very popular choice for voice conversion systems [31][62].

- Local sensitivity: A perturbation of one coefficient has only a local effect on the spectrum.
- Stability: The ascending order of the coefficients guarantees the filter stability.

- Close relationship to formants (spectral peaks): The distribution of the coefficients corresponds largely to the formant locations and bandwidths.
- Good interpolation properties.

Cepstral coefficients (MFCC, MCC): They model both spectral peaks and valleys. Being reliable for the measurement of acoustic distances they are especially useful for alignment. MCCs in particular have been widely used in both stand-alone voice conversion as well as HMM based synthesis.

Standard linear prediction coefficients give information on the formants (peaks) but not the valleys (spectral zeros) in the spectrum whereas cepstral processing treats both peaks and valleys equally. The generalized Mel-cepstral analysis method [97] provides a unification that offers flexibility to balance between them. The procedure is controlled by two parameters, α and γ, where γ balances between the cepstral and linear prediction representations and α describes the frequency resolution of the spectrum. Mel-cepstral coefficients (MCCs) ($\gamma = 0$, $\alpha = 0.42$ for 16 kHz speech) are a widely used representation in both voice conversion and HMM-based speech synthesis [98][99][100][101].

Spectral samples: Spectrum samples have been used typically in frequency warping based systems.

In addition to the spectral envelope parameterizations introduced previously some typical features related to the source excitation are *pitch* and *voicing*. Sometimes the details of the *excitation spectra* need to be modeled as well, as is the case for sinusoidal modeling. A common approach for pitch conversion is to shift the mean and scale the variance of F_0 or $\log(F_0)$ to the values of the target speaker. The *voicing* feature describes the level of aperiodicity and can take various forms from a simple binary variable to more refined representations which allow various degrees of aperiodicity or describe the aperiodicity for individual frequency bands.

2.3 ALIGNMENT TECHNIQUES

Voice conversion systems learn a mapping function based on a correspondence between the source and target training data. This correspondence is established in a process called alignment and can be realized in several ways. In the most typical case it is realized between acoustic classes or between individual frames but an alignment in the strict sense may also be omitted using adaptation techniques to adapt an already trained transformation. The specific alignment strategy and method are largely influenced by the properties of the training corpus and by the concrete spectral transformation used.

The characteristics of the training corpus lead to several voice conversion scenarios. The training data is said to be *parallel* or *text-dependent* if the utterances of the source and target speakers are based on the same text and it is called *non-parallel* or *text independent* otherwise. An extreme case of text independent voice conversion is the *cross-lingual* conversion where the speakers speak different languages with typically different phoneme sets. For parallel data, well-studied methods are able to achieve optimal time-alignment while for the other cases the source-target correspondence is not always obvious. The main alignment paradigms are discussed in the following paragraphs.

2.3.1 Frame-to-Frame Alignment

The most common type of alignment is in the form of source-target frame pairs. This frame-level correspondence can be realized in many ways depending on the type of training data. The parallel data has the advantage that exactly the same sentences are uttered by the source and target speakers which guarantees that the phonetic sequences are the same for both speakers. In this case the dominant alignment technique is dynamic time warping (DTW) [26][25][31][54][102] which determines the optimal source-target pairs by searching the path of minimal global distortion. The main shortcoming of DTW is that it doesn't take into account the differences between speakers. For best results speaker normalization should precede DTW but this is typically omitted since the normalization itself requires a predetermined source-target correspondence. Stylianou proposed a method [103] in which the result of a first DTW alignment is used to obtain an initial estimate of the conversion function while a second DTW realigns the converted and target vectors increasing the alignment accuracy.

Another possibility to align parallel training utterances, if the phonetic transcriptions are known, is based on HMMs. The sentences are segmented into phonetic units using speaker dependent models and linear time warping [104] or dynamic time warping [56] is applied inside the units to obtain a high-accuracy correspondence between the source and target vectors. A disadvantage of this approach is that the training of accurate speaker dependent HMMs requires rather large data from both speakers.

In [105] statistical methods and maximum likelihood criteria have been used to optimize simultaneously the spectral conversion function and the correspondence of vector sequences. This approach is reported to outperform the typical case when alignment and training are separate processes.

Although parallel training data simplifies the alignment, in a more realistic scenario only non-parallel corpora may be available. A number of methods have been proposed to determine the source-target frame correspondence from such non-parallel data. Some of these ideas are specifically addressed to intra-lingual voice conversion while others, based exclusively on acoustic features, are also compatible with the cross-lingual case.

The customization of a TTS synthesizer is a special application where the alignment problem can be reduced to the parallel case by using the TTS system to generate the same sentences as those of the target speaker [104]. This approach is not compatible with cross-lingual applications.

In a technique suggested by [106] and developed as an intra-lingual solution, state indexes are assigned to all source and target frames using a speech recognizer with speaker-independent HMMs. Given a state sequence of the source speaker the method retrieves the longest matching sub-sequences from the target speaker labeled data until all the frames in the source sequence are paired.

In cross-lingual voice conversion it is impossible to obtain parallel training corpora because the phoneme sets of the two speakers are different. This is the most extreme case in terms of alignment but once the alignment is solved the training of conversion functions becomes an easy task. When at least one of the speakers is bilingual it is possible to obtain a parallel corpus and use conventional methods to estimate the conversion function [107][108][104]. In practice the requirement of bilingual speakers represents a serious limitation. For aligning frames over a cross-lingual corpus it is desirable to use techniques based exclusively on acoustic features for several reasons: they avoid the need for bilingual speakers and since they do not use any linguistic information they allow text-independent and language-independent voice conversion. Moreover, these techniques transform the non-parallel

and cross-lingual corpora into parallel ones. A selection of methods compatible with both non-parallel and cross-lingual training data is presented next.

Suendermann [109] proposed a class mapping approach. The source and target data are clustered independently and each source acoustic class is mapped to a target class based on the similarity of associated vocal-tract-normalized centroids. All the vectors in all the classes are mean-normalized and the alignment of each source frame is realized by finding the nearest neighbor in the corresponding target class.

Given a source vector sequence $\{s_k\}$ of length N a dynamic programming technique was proposed [110] to find a corresponding sequence of target vectors $\{t_k\}$ that minimizes the cost function:

$$C(\{t_k\}) = \alpha \sum_{k=1}^{N} d(s_k, t_k) + (1-\alpha) \sum_{k=2}^{N} d(t_k, t_{k-1}) \tag{2.12}$$

where the function $d(.)$ represents and acoustic distance between two spectral vectors. The cost function resembles the idea of unit selection being composed of a source-target cost (first term) and a concatenation cost (second term) whose relative importance can be adjusted though the parameter α. The downsides of this method are the intensive computation but also the fact that the optimal sequence $\{t_k\}$ becomes closer to $\{s_k\}$ when the training data is increased leading to poorer conversion results.

In [111] the authors present an alignment scheme for text-independent voice conversion maintaining phonetic accuracy and the internal topology of the parameter space. The common phonetic clusters are used to determine an initial weighted linear mapping. The mapping is based on several of the nearest phonetic clusters in order to ensure continuity. The result of this alignment is used to initialize a self-organizing iterative learning algorithm for nonlinear data alignment. The self-organizing algorithm learns a topology-preserving mapping in which neighboring input points are mapped to neighboring outputs. For the cross-lingual case a manifold expansion scheme is used to map the unaligned source vectors according to the aligned data starting with those source vectors which have a maximum number of aligned neighbors. The evaluations show improved alignment accuracy and stability.

The approach introduced in [112] is a data-driven alignment method for nonparallel corpora based on the iteration of some existing voice conversion techniques. The alignment is initialized with a nearest neighbor scheme which maps each source vector to the nearest target vector and each target vector to the nearest source vector. This data is used to train a GMM and the traditional GMM conversion is applied to the source data. The nearest neighbor alignment is repeated replacing the source vectors with the converted ones. By iterating these steps the converted vectors are converging towards the corresponding target vectors leading to a progressive improvement of the alignment. The method does not require phonetic information and is shown to achieve similar performance to an equivalent system trained on a parallel corpus.

2.3.2 Alignment of Acoustic Classes

In some voice conversion systems the alignment is realized as a correspondence between acoustic classes and the conversion functions are typically class dependent. The systems based on codebook mapping or frequency warping [52] [113] are good examples. A common classification of the acoustic data is based on phonemes or phoneme groups.

In [113] the acoustic classes are defined as states of a HMM. A speaker-independent HMM is used to segment the source and target speakers' utterances into states establishing an implicit correspondence between them. In [52] the classification is realized using clustering and the correspondence between source and target classes is determined based on minimum-distance criteria.

2.3.3 No Alignment

An alignment in the strict sense is not always necessary. One possibility is to train an acoustic model for one of the speakers and use the model to determine an optimal transformation function. For example, in [56], hidden Markov models of the target speaker are trained and the conversion function is estimated such as to maximize the likelihood of the converted source vectors with respect to the target models. The conversion function used there is based on a Gaussian mixture model of the source space.

In some other cases an already trained transformation between speakers A and B can be adapted to the acoustic data of a new target speaker C. There are two different adaptation techniques: maximum-a-posteriori (MAP) adaptation [114] and maximum-likelihood stochastic transformations (MLST) [115]. With MLST the conversion function can also be adapted to a different source speaker.

Another case where the alignment is not necessary is the voice transformation obtained through adaptation techniques and HMM-based speech synthesis. In this case an initial HMM estimated from the training data of the source speaker is adapted to maximize the likelihood of the target speaker's data with respect to the modified HMM [116][117].

2.4 SPECTRAL CONVERSION

In general, the construction of a voice conversion system starts from an analysis/synthesis technique which allows the reconstruction of speech signal with minimum loss of quality. The speech representation and type of parameterization are chosen depending on the desired modifications and algorithms used. The spectral modification aims to transform the spectral/acoustical features preserving the speech quality and represents a central task of voice conversion systems. The spectral information is typically manipulated in the form of spectral envelopes which are smoothed versions of the Fourier magnitude spectra, often independent of the fundamental frequency. Most part of voice conversion literature is concerned with the spectral conversion task and numerous solutions have been proposed. Despite their diversity these techniques can be grouped in several categories depending on the type of transformation. This section presents in more detail the existing methods and algorithms for spectral conversion.

2.4.1 Mapping Codebooks

A basic voice conversion technique is codebook mapping [26]. The simplest way to realize codebook based mapping would be to train a codebook of combined feature vectors z. Then, during conversion, the source side of the vectors could be used for finding the closest codebook entry, and the target side of the selected entry could be used as the converted vector. The classical paper on codebook based conversion [26] proposes a slightly different approach that can utilize existing vector quantizers. There the training phase involves generating histograms of the vector correspondences between the quantized and aligned source and target vectors. These histograms are then used as weighting

functions for generating a linear combination based mapping codebook. Regardless of the details of the implementation, codebook based mapping offers a very simple and straightforward approach that can capture the speaker identity quite well, but the result suffers from frame-to-frame discontinuities and poor prediction capability on new data.

A fundamental problem of codebook mapping is the discrete representation of the acoustic spaces as a limited set of spectral envelopes. Another severe problem is caused by the frame-based operation which ignores the relationships between neighboring frames or any information related to the temporal evolution of the parameters. These facts produce spectral discontinuities and lead to a degraded quality of the converted speech. A number of methods have been proposed in the literature to address these drawbacks and improve the spectral continuity of the codebook mapping.

In terms of spectral mapping, though, the codebook has the attractive property of preserving details of the training data. This property has been used by some authors to reinforce spectral details. For example, [118] proposed a hybrid GMM-codebook to reduce the spectral smoothing (absence of spectral details) of the GMM based conversion and the discontinuity issues of the codebook approach. Some enhancements to the basic codebook based methods are presented next in this section.

Enhancements of the mapping codebooks technique

The *STASC* (speaker transformation algorithm using segmental codebooks) method [51] addresses the problem of discrete representation of the acoustic space by utilizing a weighted sum of codewords in order to cover well the acoustic space of the target speaker. Phoneme centroids are computed for both the source and the target speaker, forming two codebooks of spectral vectors with one-to-one correspondence. In order to convert a source vector, a set of weights is determined depending on a similarity measure between the source vector and the set of centroids in the source codebook. The conversion is realized by using the weights to linearly combine the corresponding centroids in the target codebook. While improving the continuity with respect to the basic codebook approach, this method causes severe over-smoothing by summing over a wide range of different spectral envelopes.

A later version of the STASC method [55] introduces a number of refinements. First, it eliminates the aligned source and target classes if they are significantly different due to accent reasons. Spectral equalization is used to compensate for the differences in recording environments and a pre-emphasis filter is proposed to increase the robustness to small variations in the speech signal.

In [119] the authors noticed that, for a better detail preservation, only similar envelopes should be included in the averaging process and proposed a *phoneme tied weighting* scheme which splits the codebook into groups by phoneme types. The weighted summation is applied only to code-words belonging to the same group.

Hierarchical codebook mapping [120] aims to improve the precision of the spectral conversion by estimating and adding a residual term to the typical codeword mapping. In addition to the mapping codebook between the source vectors x and the target vectors y, a new codebook is trained from the same source vectors x and the corresponding conversion residuals $\varepsilon = y - \hat{y}$. The residuals represent the differences between a real target vector y aligned to x and x's conversion through the first codebook, $\hat{y}$. In conversion, both codebooks are used; the first for predicting a target codeword $\hat{y}$ and the second to find the corresponding residual ε. The final result of the conversion is obtained by summing outputs of the two codebooks, i.e. $\hat{y}' = \hat{y} + \varepsilon$. Although hierarchical codebook mapping

improves to some extent the precision compared to the basic codebook based conversion, this approach is essentially only producing a finer representation of the acoustic space while being otherwise likely to inherit the fundamental problems of the basic codebook mapping.

Trellis structured vector quantization [121] tackles the problem of discontinuities common for many codebook-based conversion approaches. The method operates with blocks of consecutive frames to obtain dynamic information and uses a trellis structure and dynamic programming to optimize a codeword path based on this dynamic information. Parallel training speech quantized in the form of codeword sequences is aligned and source-target codeword pairs are formed. Preceding codewords in the source and target sequences are combined with each pair forming blocks of consecutive codewords which reflect the speech dynamics. The conversion of a source speech sequence requires the construction of an equally long trellis structure whose lines correspond to the codewords of the target codebook. The nodes in the trellis structure are assigned an initial cost and a maximum number of so called survivor paths, or valid preceding target codewords. The initial cost is based on the similarities between the consecutive frames from the input sequence and memorized blocks consecutive source codewords while the survivor paths are selected based on memorized blocks of consecutive target codewords. The survivor paths are also associated a transition cost based on Euclidean distance. Dynamic programming is used to find the optimal path in the trellis structure resulting in a converted sequence of target codewords. The method proposes a rigorous way to handle the spectral continuity by utilizing dynamic information and keeping at the same time the advantages of good preservation of spectral details provided by the codebook framework. The approach was shown to clearly outperform the basic GMM and codebook-based techniques which are known to suffer from over-smoothing and discontinuities respectively. Compared to conventional VQ-based methods, the additional cost of this approach is an increase of the training model size by 25% and extra computation related to the trellis algorithm during the conversion phase.

A similar idea based on *dynamic programming* is found in [122]. Parallel training utterances are first segmented into phonemes using HMMs and the frame alignment is obtained by applying DTW inside each phone. The data of the two speakers is vector-quantized into separate codebooks $\{S_i\}_{i=1..L}$ and $\{T_j\}_{j=1..L}$ and an LxL histogram matrix H is obtained from the sequence of aligned and quantized training vectors where $H(i,j)$ represents the number of occurrences of the pair $\{S_i, T_j\}$. Another $L \times L$ matrix P is obtained for the target speaker, in which $P(i,j)$ represents the transition probability from class i to class j, based on the occurrences found in the training data. A source utterance given for conversion is first represented as a vector-quantized sequence of length N. An associated NxL histogram matrix $H^{(sen)}$ is built such that the k-th row of $H^{(sen)}$ is the row in H corresponding to the k-th frame, representing the probabilities of each target class to form a pair with it. Finally, the best path from the first row of $H^{(sen)}$ to the last one is found through dynamic programming. The result is a length N sequence of target codewords that maximizes the product between node probabilities in $H^{(sen)}$ and transition probabilities obtained from P included in the path.

2.4.2 GMM and Linear Transformations

Voice conversion based on *linear multivariate regression* was introduced in [22]. The parameter vectors of the source speakers are vector-quantized into Q classes and for each class a transformation matrix T_q is determined as a least square solution to:

$$C_q^{(t)} = T_q C_q^{(s)} \tag{2.13}$$

where the columns of matrix $C_q^{(s)}$ are mean-normalized source vectors in class q and those of $C_q^{(t)}$ are formed by their aligned mean-normalized target vectors. In order to convert a sequence of source vectors, the frames are first vector-quantized and then normalized and converted according to the mean vector and transformation matrix of the assigned class. This method based on linear transformations achieves a good similarity to the target voice but the quality is affected by some audible distortions.

The emergence of statistical methods marked an important progress in the field of voice conversion. A major drawback of the systems based on vector-quantization is related to the transition between classes which may produce discontinuities. This can be avoided if the acoustic space is divided into overlapping classes to which input vectors belong with a certain probability. The idea proposed by Stylianou in [19][25] is to fit a *Gaussian mixture model* (GMM) to the training vectors of the source speaker using the expectation-maximization (EM) algorithm [123].

$$p(x) = \sum_m \alpha_m N(x, \mu_m, \Sigma_m) \tag{2.14}$$

In this model of the source speaker's acoustic space, $N(x, \mu_m, \Sigma_m)$ are Gaussian distributions defined by their mean vectors μ_m and covariance matrices Σ_m, and α_m are weights of these Gaussian components; $p(x)$ denotes the probability distribution function (pdf) of the acoustic vectors x of the source speaker. The conversion function is defined as:

$$F(x) = \sum_m p_m(x)[v_m + \Gamma_m \Sigma_m^{-1}(x - \mu_m)] \tag{2.15}$$

where $p_m(x)$ denotes the conditional probability of the m^{th} Gaussian component given the observation x and the parameters v_m and Γ_m are calculated by minimizing the squared error between the transformed vectors $F(x_t)$ and their aligned target vectors y_t over a parallel training corpus.

A GMM can also be used to approximate a joint density of the source and target features. In [31][124], the training vectors x of the source speaker and their aligned target pairs y are concatenated and a *joint density GMM* is trained from the resulting vectors $z = [x^T, y^T]^T$ using the EM algorithm [123]. The density of the joint vectors z is modeled as:

$$P(z) = P(x, y) = \sum_{m=1}^{M} \alpha_m N(z, \mu_m, \Sigma_m) \tag{2.16}$$

where α_m is the prior probability of the m^{th} Gaussian component $N(z, \mu_m, \Sigma_m)$, M denotes the number of components, and the mean μ_m and covariance Σ_m of the m^{th} component are further expressed as:

$$\mu_m = \begin{bmatrix} \mu_m^{(x)} \\ \mu_m^{(y)} \end{bmatrix}, \quad \Sigma_m = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix}. \tag{2.17}$$

The conditional probability of a converted vector y obtained from the input vector x through the mth Gaussian component is a Gaussian distribution defined by the mean $E_m^{(y)}$ and the covariance $D_m^{(y)}$:

$$\begin{aligned} E_m^{(y)} &= \mu_m^{(y)} + \Sigma_m^{(yx)} \left(\Sigma_m^{(xx)}\right)^{-1} \left(x - \mu_m^{(x)}\right), \\ D_m^{(y)} &= \Sigma_m^{(yy)} - \Sigma_m^{(yx)} \left(\Sigma_m^{(xx)}\right)^{-1} \Sigma_m^{(xy)}. \end{aligned} \tag{2.18}$$

The conversion function which minimizes the mean squared error

$$\varepsilon_{mse} = E[\|y - F(x)\|^2], \tag{2.19}$$

is again a weighted sum of local regressions determined from the model parameters (α, μ, Σ).

$$F(x) = E[y|x] = \int P(y|x,\lambda) \cdot y \cdot dy = \int \sum_{m=1}^{M} P(m|x,\lambda) \cdot P(y|x,m,\lambda) \cdot y \cdot dy$$
$$= \sum_{m=1}^{M} P(m|x,\lambda) \cdot E_m^{(y)} = \sum_{m=1}^{M} p_m(x) \cdot \left(\mu_m^{(y)} + \Sigma_m^{(yx)} \left(\Sigma_m^{(xx)} \right)^{-1} \left(x - \mu_m^{(x)} \right) \right) \quad (2.20)$$

Here $E[.]$ denotes expectation, $P(.)$ probability density function, λ is the GMM model, and $p_m(x) = P(m|x,\lambda)$ represents the posterior probability of the m^{th} Gaussian component for the observation x:

$$p_m(x) = \frac{\alpha_m \cdot N\left(x, \mu_m^{(x)}, \Sigma_m^{(xx)}\right)}{\sum_{l=1}^{M} \alpha_l \cdot N\left(x, \mu_l^{(x)}, \Sigma_l^{(xx)}\right)} \quad . \quad (2.21)$$

The GMM based methods achieve a balance between quality and identity mapping which outperforms the previous techniques and they have become the most popular voice conversion approach. The soft classification based on GMM has the role to eliminate the artifacts caused by sudden changes of the conversion function. On the other hand, the GMM based conversion has a number of shortcomings as follows.

The control of model complexity is a crucial issue when learning a model from data. There is a trade-off between two objectives: model fidelity and the generalization-capability of the model for unseen data. This trade-off problem, also referred to as bias-variance dilemma [125], is common for all model fitting tasks. In essence, simple models are subject to over-smoothing, whereas the use of complex models may result in over-fitting and thus in poor prediction ability on new data. In addition to over-smoothing and over-fitting, a major problem in conventional GMM-based conversion, as well as in many codebook based algorithms, is the time-independent mapping of features that ignores the inherent temporal correlation of speech features. Despite its problems GMM based voice conversion has been a dominating technique in the field. In the following, we discuss these problems in more detail and review some solutions proposed to overcome them.

Over-fitting

In GMM-based voice conversion the GMM may be over-fitted to the training set as demonstrated in Figure 2.3. In particular, a GMM with full covariance matrices is difficult to estimate and is subject to over-fitting [126]. With unconstrained (full) covariance matrices, the number of free parameters grows quadratically with the input dimensionality. One solution is to use diagonal covariance matrices Σ^{xx}, Σ^{xy}, Σ^{yx}, Σ^{yy} and with an increased number of components. In the joint-density GMM, this results in converting each feature dimension separately. In reality, however, the p^{th} spectral descriptor of the source may not be directly related to the p^{th} spectral descriptor of the target, making this approach inaccurate.

Over-fitting of the mapping function can be avoided by applying partial least squares (PLS) for regression estimation [99]; a source GMM (usually with diagonal covariance matrices) is trained and a mapping function is then estimated using partial least squares regression between source features weighted by posterior probability for each Gaussian and the original target features.

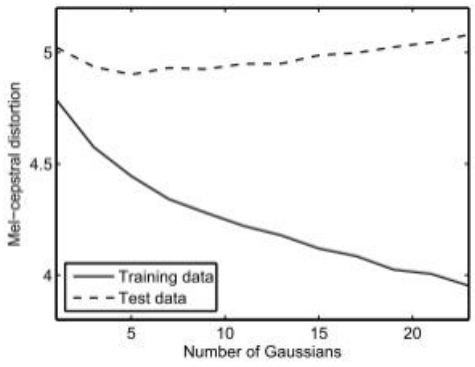

Figure 2.3: Example of over-fitting. Increasing the number of Gaussians reduces the distortion for the training data but not necessarily for a separate test set because the model might be over-fitted to the training set. (from [50])

Over-smoothing

Over-smoothing occurs both in frequency and in the time domain. In frequency domain, this results in losing fine details of the spectrum and in broadening of the formants. In speech coding, it is common to use post-filtering to emphasize the formants [127] and similarly post-filtering can also be used to improve the quality of the speech in voice conversion. It has also been found that combining the frequency warped source spectrum with the GMM-based converted spectrum reduces the effect of over-smoothing by retaining more spectral details [128].

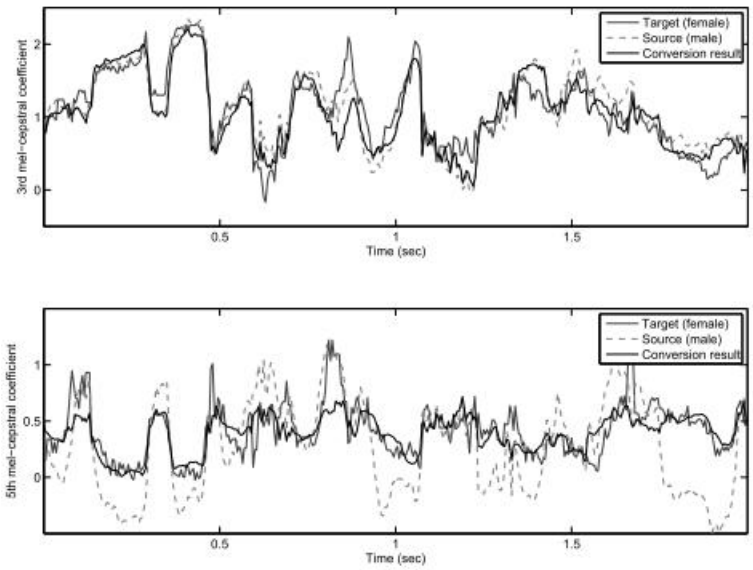

Figure 2.4: Example of over-smoothing. Linear transformation of spectral features is not able to retain all the details and causes over-smoothing. The conversion result (black line) is achieved using linear multivariate regression to convert the source speaker's MCCs (dashed gray line) to match with the target speaker's MCCs (solid gray line). (from [50])

In time domain, the converted feature trajectory has much less variation than the original target feature trajectory. This phenomenon is illustrated in Figure 2.4. According to [129], over-smoothing occurs because the term $\Sigma_m^{(yx)}(\Sigma_m^{(xx)})^{-1}$ in equation (2.20) becomes close to zero and thus the converted target becomes only a weighted sum of means of GMM components as

$$\hat{y} = \sum_{m=1}^{M} p_m(x)\mu_m^{(y)} . \quad (2.22)$$

To avoid the problem, the source GMM can be built from a larger data set and only the means are adapted using maximum a posteriori estimation [129]. Thus, the converted target becomes:

$$\hat{y} = x + \sum_{m=1}^{M} p_m(x)\left(\mu_m^{(y)} - \mu_m^{(x)}\right) . \quad (2.23)$$

Global variance can be used to compensate for the reduced variance of the converted speech feature sequence with feature trajectory estimation [100]. Alternatively, the global variance can be accounted already in the estimation of the conversion function; this degrades the objective performance but improves the subjective quality [130].

Time-independent mapping

The conventional GMM-based method converts each frame regardless of other frames and thus ignores the temporal correlation between consecutive frames. This can lead to discontinuities in feature trajectories and thus degrade perceptual speech quality. It has been shown that there is usually only a single mixture component that dominates in each frame in GMM-based voice conversion approaches [99]. This makes the conventional GMM-based approaches to shift from a soft acoustic classification method to a hard classification method, making it susceptible to discontinuities similarly as in the case of codebook based methods.

The solution proposed in [131] to the time-independency problem is to follow the GMM mapping function with a frame selection algorithm based on the principle of unit selection used in concatenative speech synthesis. The Viterbi algorithm is used to minimize an overall distance between the final sequence of target vectors and the sequence resulted through GMM mapping. In [100], the same problem is addressed introducing maximum likelihood (ML) estimation of the spectral parameter trajectory. Static source and target feature vectors are extended with first-order deltas, i.e $z = [x^T, \delta x^T, y^T, \delta y^T]^T$ and a joint-density GMM is estimated. In synthesis, both converted mean and covariance matrices of the equations (2.18) are used to generate the target trajectory. The trajectory estimation is similar to HMM-based speech synthesis. A recent approach [132] bears some similarity to [100] by using the relationship between the static and dynamic features to obtain the optimal speech sequence but does not use the transformed mean and (co)variance from the GMM-based conversion. To obtain smooth feature trajectory, the converted features can be low-pass filtered after conducting the GMM-based transformation [129] or the GMM posterior probabilities can be smoothed before making the conversion [99]. Instead of frame-wise transformation of the source spectral features, each phoneme was modeled to consist of event targets and these event targets are used as conversion features [133].

2.4.3 Frequency Warping

Frequency warping is the third basic approach to voice conversion. Given a pair of source and target spectra or spectral envelopes $X(\omega)$ and $Y(\omega)$, the technique determines a frequency warping function $w'(\omega)$ that minimizes the spectral distance between $X(w'(\omega))$ and $Y(\omega)$. The undesired effect of the glottal source can be eliminated from this process by estimating and removing the spectral tilt from X and Y before the path search. In the simplest case, the warping function can be formed based on

spectra representing a single voiced frame [27] and applied to all the spectral envelopes of a given source utterance. In the original article [22], the acoustic space is clustered using a vector-quantization procedure and a different warping function is defined for each class. In the conversion phase, the source spectra are frequency warped depending on their acoustic class. These two basic versions have inherent drawbacks related to oversimplification and hard clustering but represent the basis for a number of refinements proposed in the literature. The oversimplified use of a single warping function leads to low identity conversion scores while the hard clustering produces discontinuities. As we will see later, the frequency warping methods have some additional problems with preserving the shape of the modified formants and spectral amplitudes mapping which are difficult to address.

The problem of hard clustering is addressed in [52] with a smoothing technique applied to the parameters of the warping functions of consecutive frames. The discontinuities between classes are in this way avoided leading to a high quality of the converted speech. The less successful identity conversion is most likely caused by the poor conversion of formant amplitudes and bandwidths.

In [134][135] frequency warping is combined with bi-dimensional HMMs of formant trajectories. Initially, phonetic state segmentation is carried out by forced-alignment with speaker-dependent HMMs. Formant candidates for each HMM state are obtained by LPC analysis as poles of the LP model. The poles that do not represent real formants are discarded and the formants, represented in terms of frequency, bandwidth and amplitude, are modeled using another HMM along the frequency axis. The result is a two-dimensional HMM which models the formant trajectories. The conversion of a given source frame t is described by the formula:

$$Y[\omega, t] = \gamma(\omega, t) \cdot X[\alpha(\omega, t) * \beta(\omega, t) * \omega, t] \tag{2.24}$$

This warping function consists of a formant frequency component $\alpha(\omega, t)$, a bandwidth component $\beta(\omega, t)$ and a spectral magnitude component $\gamma(\omega, t)$. An important drawback of this method is the difficulty of estimating the formants.

Erro et al. [136] proposes a combination between a time varying frequency warping function and an energy correction filter, both derived from a GMM. A warping function is trained for each GMM component from representative source and target spectral envelopes. The warping function used to convert a given source spectral envelope is calculated as a weighted combination of the basic warping functions obtained in the training. The energy distribution of the frequency warped spectrum is modified according to the spectral envelope obtained with the classical GMM approach using an energy correction filter. The method is reported to achieve a better quality than the classical GMM approach because of the frequency warping which alleviates over-smoothing. On the other hand, it maintains the identity conversion scores similar to the GMM method due to the energy correction.

The frequency warping methods can at best obtain very high speech quality but have limitations regarding the success of identity conversion, due to problems in preserving the shape of modified spectral peaks and controlling the bandwidths of close formants. Moreover, frequency warping does not allow formant merging or splitting which is often desirable in spectral conversion [137]. Proper controlling of the formant amplitudes is also challenging. Despite numerous refinements proposed in the literature the above-mentioned fundamental problems remain largely unsolved.

2.4.4 Nonlinear Models

Artificial neural networks (ANN) represent a powerful tool for modeling complex (nonlinear) relationships between input and output. They have been applied in voice conversion almost at the same time when GMM methods were introduced but did not reach such an important impact. Their main disadvantage is the massive tuning needed to select the optimal network architecture.

In [138] artificial neural networks are used to learn a transformation of the first three formants between the source and the target speaker in a system whose speech generation is based on a formant synthesizer. A comparative study presented in [139] found the systems based on neural networks to be outperformed by GMM-based systems.

In [140] the conversion of LPC spectral envelopes is realized using a radial basis function (RBF) network with three layers. The input vector x is applied to all the RBFs in the hidden layer which are defined by radially symmetric response functions with respect to their centroids c_i:

$$h_i(x) = exp\left(-\frac{1}{2\sigma_i^2}\|x - c_i\|^2\right) \tag{2.25}$$

where σ_i represents the spread coefficient.

Each element of the output vector y is a linear combination of the responses h_i from the hidden layer:

$$y_k(x) = \sum_i h_i(x) w_{ki} \tag{2.26}$$

where $y_k(x)$ is the k^{th} element of the output vector $y(x)$, and w_{ki} are the weighting coefficients.

For a simpler training, the centroids c_i are calculated by applying the k-means algorithm to the source training vectors and σ_i is defined as $\|c_i\|^2$. The weight coefficients w_{ki} are found by minimizing the squared error of the conversion over the training data.

The voice conversion approach introduced in [98] is based on a multi-layer feed forward ANN trained from parallel data aligned with dynamic programming. The article also presents comparative results with the state of the art GMM system [100] indicating that the proposed ANN technique achieves similar performance levels. Finally, the article introduces an ANN technique able to capture speaker specific characteristics of a target speaker and to perform voice conversion without need for training data from the source speaker. In this method ANN is used to train a mapping between a linguistic descriptive parameterization to a linguistic and speaker descriptive representation of the same target data. More precisely the linguistic representation is a low order LP spectrum while a high order LP spectrum would contain both linguistic and speaker related information. Once the mapping model is trained, it can be used to convert linguistic information from any arbitrary source speaker into the target speaker voice.

Another alternative to model a nonlinear relationship is the kernel *partial least squares regression* [141]; a kernel transformation is carried out on the source data as a preprocessing step and PLS regression is applied on kernel transformed data. In addition, the kernel transformed source data of the current frame is augmented from kernel transformed source data from the previous and next frames before regression calculation. This helps in improving the accuracy of the model and maintaining the temporal continuity that is a major problem of many voice conversion algorithms.

In [142], *support vector regression* was used for spectral conversion motivated by its remarkable capability to map non-linear relationships, to find the global minimum and remain at the same time

less prone to over-fitting than the methods based on ANN or GMM. The mapping function of the multi-dimensional SVR used here is given by:

$$F(X_t) = W\phi(X_t) + b \tag{2.27}$$

where $\phi(X_t)$ is a mapping function from the P-dimensional input space to a higher dimensional space and $W = [w_1, \dots, w_P]^T$ and $b = [b_1, \dots, b_P]^T$ represent linear regressors determined by optimization techniques. The method uses a mixed kernel to balance between interpolation and extrapolation abilities improving the mapping performance. The discontinuities in the converted speech are overcome first by utilizing the dynamic information between the frames of the source speaker and secondly by adopting in the conversion phase a median filter to smooth the trajectories of the converted spectral parameters. The method is reported to achieve a good identity conversion and sound quality outperforming the state-of-the-art GMM based method. Compared to neural networks, the tuning of support vector regression is less demanding.

2.4.5 HMM Based Methods

As it was mentioned in the Introduction, a distinct branch of voice conversion has developed around the HMM-based speech synthesis using model adaptation techniques. Apart from this, there are stand-alone approaches to voice conversion based on HMMs that are not used for synthesis but for training conversion functions.

In [143], the vocal tract parameters are converted using a linear transformation:

$$v' = Av + b \tag{2.28}$$

The conversion parameters are trained from a large speech corpus of the source speaker and a small number of target utterances as follows. Speaker-dependent HMMs are trained from the source speaker's corpus and then the MLLR technique is applied to find A and b which maximize the likelihood of the target data with respect to the mean-adapted HMMs. In the conversion phase, the transformation function given by A and b is applied for the conversion of the source vectors and then the transformed speech is generated from the converted parameters.

The method proposed in [53] estimates a HMM from the training data and then a linear transformation similar to that used in GMM based conversion is calculated for each state based on the corresponding source-target frame pairs. A given source utterance is converted using state dependent conversion functions after a preliminary segmentation based on the trained HMM.

In [56], HMMs are trained only from the target speaker's data. A probabilistic transformation based on a GMM of the source vector space is determined such as to maximize the likelihood of the transformed vectors with respect to the HMM model of the target.

2.4.6 Residual Conversion

The methods reviewed previously perform the conversion of spectral envelope parameterizations derived in general by linear prediction. Some of these results considered satisfactory for the conversion of the vocal tract, in particular those based on GMMs, determined some authors to look for increasing the accuracy of the spectral transformation by transforming also the residual or excitation part of the signal, in order to improve the identity conversion scores and the quality of the converted speech. In [131], the use of the target residual (which retains a lot of the voice quality) in the synthesis of the converted speech was found to significantly increase the converted-to-target

voice similarity. The techniques discussed in this section are not performing spectral envelope conversion but act to complement it.

Since the vocal tract contribution cannot be perfectly estimated by an all-pole filter we can consider that the residual or excitation signal carries several types of information as follows:

- formants: The order of the vocal tract filter is obtained as a tradeoff between the frequency resolution and a vector length that permits a reliable transformation. Usually it is chosen as the lowest order that provides a desired quality level. Consequently, some of the spectral peaks and valleys are not captured well by this filter representation. Moreover, for certain phonemes like nasal consonants the spectral envelope contains not only poles but also zeroes. All these aspects are reflected in the residual or excitation signal.
- phase: By approximating of the spectrum as an all-pole filter it is also assumed that the minimum-phase response of the filter can model the phase envelope. This is not very realistic, therefore part of the phase information is included in the residual during the process of inverse filtering of the original speech.
- the glottal source: If the all-pole filter is determined from the samples when the glottis is closed, the residual contains useful information about the glottal source, which is reported to have a strong relationship with the emotional aspect of the speech.
- noise.

If the residual is not modified as part of the voice conversion process the resulting utterance may be perceived as coming from a third speaker. In the next paragraphs, some residual conversion methods proposed in the literature are reviewed.

In the STASC method [51] introduced previously, the conversion of the vocal tract parameter vectors is realized as a weighted summation of codewords of the target speaker in which the weights v_i are determined from the distances of the input vector to each of the source codewords. A transfer function $H(\omega)$ of the vocal tract filter is obtained by dividing the converted and input spectral envelopes. The transfer function $H_g(\omega)$ for the excitation is determined from the same weights as follows:

$$H_g(\omega) = \sum_{i=1}^{L} v_i \frac{U_i^t(\omega)}{U_i^s(\omega)}, \tag{2.29}$$

where $U_i^s(\omega)$ and $U_i^t(\omega)$ represent average excitation spectra of the source and target speakers for the i-th codeword. The global transfer function for the input frame is $H(\omega) \cdot H_g(\omega)$.

The residual selection method proposed in [62] is simple and requires storing the training data of the target speaker as pairs of vocal tract vectors $\{y_k\}$ and their associated residuals $\{r_k\}$. In the conversion phase, a residual is assigned to a converted filter $F(x)$ by finding the closest y_k from the table.

In [56], a Q-mixture GMM is trained from the target LSF vectors $\{y_k\}$ which are then translated into a length Q representation $p_k = [p_1(y_k), \dots, p_Q(y_k)]^T$ given by the probabilities of $\{y_k\}$ belonging to each of the Gaussian components. If R is the matrix of residual vectors $\{r_k\}$ found in the training data aligned column wise and the matrix P is formed column wise by $\{p_k\}$ vectors obtained from the LSF vectors $\{y_k\}$, then the least squares solution to the system:

$$R = T_r \cdot P \tag{2.30}$$

is a matrix T_r of Q columns which can be regarded as a set of codeword residuals corresponding to each of the Gaussian components of the trained GMM. In the conversion phase, a converted LSF vector $F(x)$ is first translated into a probability vector p and its residual is calculated as:

$$r = T_r \cdot p \ . \tag{2.31}$$

The authors used this method for amplitude residuals only, the phase envelope being predicted differently.

A method presented in [54] consists of two phases called residual selection (similar to the one described in [62]) and smoothing. Some of the residuals determined in the residual selection phase are found to be inadequate and responsible for audible artifacts. Especially during the voiced regions the residual sequence is not expected to have abrupt variations considering the pseudo periodical nature of the signal. This problem is addressed using a Gaussian window of variable length to smooth the sequence of residuals according to the voicing of the frame to be converted.

$$r_t' = \frac{\sum_\tau N(\tau, t, \alpha\sigma_t) \cdot r_\tau}{\sum_\tau N(\tau, t, \alpha\sigma_t)} \tag{2.32}$$

The summation is done over all the residual vectors r_τ in the residual sequence associated with the converted vocal tract vectors. The term α is a constant, σ_t denotes the voicing degree for frame t taking values between 0 and 1 and $N(x, \mu, \sigma)$ is a Gaussian distribution with mean μ and standard deviation σ. In case of a voiced frame we obtain a window which averages between several neighboring residuals, while for unvoiced frames there is virtually no local smoothing and the residuals and their phase spectra change randomly.

Another method of the same author resembles the idea of unit selection used in speech synthesis and is reported to outperform the previous one. The same table of residuals and feature vectors is built from the training data then, given a sequence of converted feature vectors $\{F(x_t)\}$, the sequence of residuals $\{r_t\}$ is determined by minimizing the global cost function:

$$C(\{r_t\}) = \sum_t C_{rv}(r_t, F(x_t)) + \sum_t C_{rr}(r_t, r_{t-1}) \tag{2.33}$$

The first term is the distance between r_t and the residual assigned to $F(x_t)$ by the selection technique described in [62] and the second term can be regarded as a concatenation cost between two residuals.

In [144], the VTLN technique originally used for spectral envelopes was also applied to residuals. Despite expectations this was found to positively influence the identity conversion and have negligible effects on the quality.

A study presented in [145] revealed that there is a much higher correlation between the vocal tract filter and excitation of the same speaker than it is between the residuals of different speakers. The authors compared three schemes which used the same technique for vocal tract conversion but different methods of residual manipulation as follows:

- No residual conversion, the source residual is left unchanged.
- Predicting the target residual from the input source residual based on a codebook of aligned source and target residuals.
- Predicting the target residual from the converted vocal tract based on a similar codebook.

The study showed that the latter method produced the best results.

The previous finding is taken one step further in [146] where the residual is considered to depend not only on the vocal tract but also on f_0. The authors report performance improvements due to the sub-classification based on f_0.

In contrast to other methods which cluster the training LSF vectors $\{y_n\}$ of the target speaker, the method presented in [147] clusters their associated residuals $\{r_n\}$ instead. The distribution of LSF vectors is then modeled by independent GMMs for each cluster and the transition probability between the clusters is extracted from the training data. In the conversion phase, the residual r associated to a converted LSF vector $F(x)$ is found as a linear combination of the residual centroids with weights calculated from the GMMs and from the transition probabilities. This technique is reported to outperform the conventional residual prediction approaches.

The research on residual conversion appeared as a means to enhance the spectral conversion accuracy after the GMM-based vocal tract conversion has been considered successful enough by some researchers. Despite the improvements achieved by these methods, high-resolution voice conversion remains an open topic of research.

Chapter 3

Parametric Framework for Voice Conversion

The speech model used for the analysis of input signals and for the reconstruction of the modified signals represents an essential component of the voice conversion system and should be designed in accordance with the desired speech transformations. For the purpose of voice conversion a good speech model should allow flexible spectral and prosodic modifications and provide high-quality waveform re-synthesis. An additional goal is to have a voice conversion system that is compatible with TTS and with speech coding which would facilitate its use for communications related and embedded applications.

The voice conversion framework described in this chapter has been developed around a speech model used in speech coding. This leads to an efficient speech representation which, in addition, supports flexible voice manipulation. Since the same speech parameterization is used as internal representation in an existing concatenative TTS system, the voice transformations could be directly applied on the output parameter vectors of the TTS. A couple of techniques related intrinsically to the framework are also presented.

In section 3.1 the parametric speech model is introduced. Section 3.2 presents a discussion about the undesired effects that voice conversion may have on the voicing level and how to adjust the voicing parameter in line with the spectral transformation. A practical way to attenuate the noise introduced in transformation and improve the perceptual quality of the result is presented in section 3.3. Section 3.4 presents some concluding remarks and proposes for future work a technique for automatic collection of voice conversion data based on this framework. The chapter is based on work published in [36] [37] [38] [39].

3.1 VLBR CODEC SPEECH PARAMETERIZATION

Concerning the domain of conversion, many different methods have been reported in the literature, with the most common approach being a separate conversion of the vocal tract contribution and the excitation, as proposed e.g. in [26]. There are also several different proposals for the modeling of the

excitation signal in voice conversion. For example, in [26] the excitation was modeled in a very simple manner using only pitch and energy parameters, in [124] the source excitation was used with only pitch modification, while [25] used a more sophisticated harmonic + noise model.

In this section, a parametric speech model for voice conversion is introduced. The parametric representation separates the speech signal into a vocal tract contribution estimated using linear prediction and an excitation signal modeled using a scheme based on sinusoidal modeling. This parametric framework is in line with the theory of human speech production and was inspired by the successful usage of a similar speech model in a low-bit-rate speech coding application, presented earlier in [148]. The parametric model contains favorable properties from the viewpoint of both voice conversion and speech coding, and allows a seamless combination of these two aspects. An initial version of the proposed voice conversion scheme has been implemented and evaluated in listening tests. The results show that the proposed approach offers a promising framework for voice conversion although further development work is still needed to reach its full potential.

3.1.1 Parametric Speech Model

Speech representation

The speech model discussed in this section is based on the fact that a speech signal, or alternatively a vocal tract excitation signal, can be represented as a sum of sine waves of arbitrary amplitudes, frequencies and phases [24][149]:

$$s(t) = Re \sum_{m=1}^{L(t)} a_m(t)\, exp\left(j\left[\int_0^t \omega_m(t)dt + \theta_m\right]\right), \tag{3.1}$$

where, for the m^{th} sinusoidal component, $a_m(t)$ and $\omega_m(t)$ represent the amplitude and the frequency, and θ_m represents a phase offset. To obtain a frame-wise representation, the parameters are assumed to be constant over the analysis frame. Consequently, the discrete signal $s(n)$ in a given frame can be approximated as

$$s(n) = \sum_{m=1}^{L} A_m \cos(n\omega_m + \theta_m), \tag{3.2}$$

where A_m and θ_m represent the amplitude and the phase of each sine-wave component associated with the frequency track ω_m, and L denotes the number of sine-wave components.

To simplify the representation, we have assumed that the sinusoids are always harmonically related, i.e. that the frequencies of the sinusoids are integer multiples of the fundamental frequency ω_0. During voiced speech, ω_0 corresponds directly to the pitch associated with the analysis frame. During unvoiced speech, however, there is no physically meaningful pitch available, and we use a fixed value for ω_0. The voicing can be taken into account in different ways. One alternative would be to model only the voiced contribution using equation (3.2) and the unvoiced contribution could be modeled separately as a spectrally shaped noise. To further simplify the model, we assume that the sinusoids can be classified as continuous or random-phase sinusoids. The continuous sinusoids represent voiced speech and they are modeled using a linearly evolving phase. The random-phase sinusoids, on the other hand, represent unvoiced noise-like speech that is modeled using a random phase. This leads to the model:

$$s(n) = \sum_{m=1}^{M} A_m(\nu_m \cos(n\omega_m + \theta_m^V) + (1 - \nu_m)\cos(n\omega_m + \theta_m^U)), \quad (3.3)$$

where ν_m is the degree of voicing for the m^{th} sinusoidal component ranging from 0 to 1, while θ_m^V and θ_m^U denote the phase of the m^{th} voiced and unvoiced sine-wave component, respectively.

To facilitate both voice conversion and speech coding, the sinusoidal model described above is applied to the modeling of the vocal tract excitation signal. The excitation signal is obtained using the well-known linear prediction approach. In other words, the vocal tract contribution is captured by the linear prediction analysis filter $A(z)$ and the synthesis filter $1/A(z)$, while the excitation signal is obtained by filtering the input signal $x(t)$ using the linear prediction analysis filter $A(z)$ as

$$s(t) = x(t) - \sum_{j=1}^{N} a_j x(t-j), \quad (3.4)$$

where N denotes the order of the linear prediction filter. The linear prediction coefficients $\{a_j\}$ are generally estimated using the autocorrelation method or the covariance method, with the former being more popular due to the ensured filter stability. In addition to the separation into the vocal tract model and the excitation model, the overall gain or energy is used as a separate parameter to simplify the processing of the spectral information.

Parameter estimation

As described above, the speech representation used in the voice conversion system consists of three elements: i) vocal tract contribution modeled using linear prediction, ii) overall gain/energy, iii) normalized excitation spectrum. The latter is further represented using the pitch, the amplitudes of the sinusoids, and voicing information. Each of these parameters is estimated at 10-ms intervals from an 8-kHz input speech signal. Next, the parameter estimation process is described at a general level.

The coefficients of the linear prediction filter are estimated using the autocorrelation method and the well-known Levinson-Durbin algorithm, together with mild bandwidth expansion. This approach ensures that the resulting filters are always stable. Each analysis frame consists of a 25-ms speech segment, windowed using a Hamming window. The degree of the linear prediction filter is set to 10 for 8-kHz speech. For further processing, the linear prediction coefficients are converted into the line spectral frequency (LSF) representation. From the viewpoint of voice conversion, this widely-used representation is very convenient since it has a close relation to formant locations and bandwidths, and it offers favorable properties for different types of processing and guarantees filter stability.

The operation of the pitch estimation algorithm can be summarized as follows. First, a frequency-domain metric is computed using a sinusoidal speech model matching approach that partially follows the ideas presented in [150]. Then, a time-domain metric measuring the similarity between successive pitch cycles is computed for a fixed number of pitch candidates that received the best frequency-domain scores. The actual pitch estimate is obtained using the two metrics together with a pitch tracking algorithm that considers a fixed number of potential pitch candidates for each analysis frame. As a final step, the obtained pitch estimate is further refined using a sinusoidal speech model matching based technique to achieve better than one-sample accuracy.

Once the final refined pitch value has been estimated, the parameters related to the residual spectrum can be extracted. For these parameters, the estimation is performed in the frequency domain after applying variable-length windowing and fast Fourier transform (FFT). The voicing information

is first derived for the residual spectrum through the analysis of voicing-specific spectral properties separately at each harmonic frequency. The spectral harmonic amplitude values are then computed from the FFT spectrum. Each FFT bin is associated with the harmonic frequency closest to it.

The final step in the parameter estimation process is to obtain the energy value. This estimation is performed in time domain, using the root mean square energy. Since the frame-wise energy varies significantly depending on how many pitch peaks are located inside the frame, the estimation computes the energy of a pitch-cycle length signal instead.

3.1.2 Parametric Conversion Scheme

The proposed voice conversion system performs the conversion using the parametric representation presented in the previous subsection. Given a parallel training corpus, the frame alignment is obtained by HMM based phonetic segmentation and DTW following a procedure [56] introduced in section 2.3. The DTW is applied on Bark-scaled LSF vectors within one phoneme segment at a time. Non-simultaneous silent segments are disregarded. The DTW algorithm results in a combination of aligned source and target vectors $z = [x^T\ y^T]^T$ that can be used to train a conversion model. The conversion scheme adopted here follows the popular approach proposed in [124] and described in subsection 2.4.2 which uses the aligned data z to estimate the GMM parameters (α, μ, Σ) of the joint distribution $p(x, y)$. The conversion utilizes several modes, each containing its own GMM model with 8 mixture components. The modes are achieved by clustering the LSF data in a data-driven manner. More details on the mode based conversion are given in section 5.2.

Among the speech parameters described earlier in this section, pitch and LSFs were found particularly important from the perception point of view in voice conversion. In the development of our current system, the emphasis was placed on the conversion of these features. Other features such as voicing and residual spectrum were used as complementary information and were exploited in the model training but no explicit conversion was performed for these parameters in the current system.

The conversion of the LSF vectors is performed using an extended vector that also contains the derivative of the LSF vector, to take some dynamic context information into account. This combined feature vector is transformed through GMM modeling, using equation (2.20) from subsection 2.4.2. Only the true LSF part is retained after conversion.

The pitch parameter is transformed through the associated GMM in frequency domain using equation (2.20) from subsection 2.4.2. During unvoiced parts, “pitch” is left unchanged. The 8-mixture GMM used for pitch conversion is trained on aligned data with the same type of voicing for the source and target speakers.

After the conversion of the pitch parameter, the residual amplitude spectrum is processed accordingly. The reason for this processing is the fact that the length of the amplitude spectrum vector depends on the pitch value at the corresponding time instant. This means that the residual spectrum, although essentially unchanged, will be re-sampled to fit the dimension dictated by the converted pitch at that time.

After the features have been converted, they are used together to re-synthesize the transformed waveform. The synthesis is performed in a pitch-synchronous manner.

3.1.3 Evaluation Results

The parametric voice conversion system described in this section was evaluated in listening tests in the context of the second TC-Star evaluation campaign. The evaluation covered aspects related to both speaker identity and speech quality. The evaluation was carried out by an independent evaluation agency.

Test set-up

The data set [151] used in the testing included UK English speech data from four different speakers (two female and two male speakers). The training set included 159 sentences per speaker and a distinct testing set consisted of 9 sentences per speaker. The same sentences were recorded from all the speakers.

Among the 12 possible conversion directions, 4 were chosen as the directions included in the test. For the selected directions, the test organizer provided the recorded source sentences used in the test. These source sentences were converted using our voice conversion system to the voices of the target speakers. The converted signals were evaluated by 20 native non-expert listeners.

The listening test included two parts. In the first part, the listeners were asked to evaluate the speaker identity without considering the speech quality using the 5-level scale summarized in Table 3.1. The true target signals recorded from the target speakers, available only for the test organizer, were used as the reference. The listeners had to evaluate whether two given samples (converted and target) were spoken by the same person or not. In the second part, the listeners evaluated the perceptual quality of the converted speech using the mean opinion score (MOS) grades shown in Table 3.1.

Table 3.1: Scale used for the evaluation of speaker identity and speech quality.

Grade	Meaning (speaker identity)	Meaning (speech quality)
5	Definitely identical	Excellent
4	Probably identical	Good
3	Not sure	Fair
2	Probably different	Poor
1	Definitely different	Bad

Results

The results are summarized in Table 3.2 and Table 3.3. Table 3.2 contains the results from the first part of the listening test, focusing on the evaluation of speaker identity. The combined score for all the directions, not shown in the table, was 2.53. The results from the speech quality evaluation are summarized in Table 3.3.

Table 3.2: Results from the first part of the evaluation (speaker identity). F denotes a female and M a male speaker.

Direction	$F_1 \rightarrow F_2$	$F_1 \rightarrow M_2$	$M_1 \rightarrow F_2$	$M_1 \rightarrow M_2$
Score	3.10	3.05	2.20	1.77

Table 3.3: Results achieved from the second part of the evaluation (speech quality).

	MOS score
Achieved score	2.09
Reference 1 (source)	4.80
Reference 2 (target)	4.78

Discussion on the results

Based on the results, the first observation that can be made is that there were large differences between the different conversion directions. Moreover, despite the moderate average scores, the person identity conversion was sometimes perceived very successful. This can be regarded as a good result for two reasons. First, our initial system that participated in the evaluation only converted the LSFs and the pitch parameter. Moreover, the conversion was performed in a frame-wise manner without considering the frame- to-frame evolution of the parameters or intonation contours. Significant improvements can be expected after making the system more complete.

As can be seen from Table 3.3, a rather low score was achieved in the speech quality evaluation. There are a couple of clear reasons for this. First, the system produced 8-kHz output signals while the other signals (e.g. the reference samples) included in the listening test used a sampling rate of 16 kHz. Second, the source signals also contained some non-speech elements such as audible breathing and the parametric speech and conversion models turned them into audible artifacts in the output signals. Third, the frame-by-frame conversion made the converted parameter contours a bit noisy and this was also audible in the output signals. Finally, the fact that not all the parameters were converted also had its impact on the quality. Considering these underlying reasons for quality degradations, it is evident that much better quality can be expected after further development work.

3.2 VOICING LEVEL CONTROL

Speech processing related changes in the speech spectra may often lead to unwanted changes in the effective degree of voicing, which in turn may degrade the speech quality. This phenomenon is studied more closely in this section, first on a theoretical level and then in the context of voice conversion. Moreover, a simple but efficient approach for avoiding the unwanted changes in the effective level of voicing is proposed. The usefulness of the proposed voicing level control is demonstrated in a practical voice conversion system. The compensation of the changes in the degree of voicing is found to reduce the average level of noise in the output and to enhance the perceptual speech quality.

The concept of voicing is one of the fundamental concepts in speech processing. Roughly speaking, from the speech production point of view, voiced sounds are produced by forcing air through the glottis with the tension of the vocal cords adjusted so that they vibrate in oscillation. Unvoiced sounds, on the other hand, are produced without the oscillation of the vocal cords. In human speech, vowels are usually considered voiced while consonants can be voiced or unvoiced. Voicing related information can be utilized in many ways in speech processing.

In some very simplified speech models, such as in the basic source-filter model used in the well-known linear prediction coding (LPC) based LPC-10 vocoder [152], all segments of speech are regarded as either fully voiced or fully unvoiced. The voiced excitation is modeled as periodic pulses

while the unvoiced excitation is represented using random noise. Even though this approach is quite well in line with the simplified view on human speech production, it has been found inadequate for producing high quality speech. A natural way for improving the performance is to allow different degrees of voicing in the excitation modeling. There are many successful speech models that utilize this idea in different ways. For example, the model used in waveform interpolation (WI) speech coding [153] allows voiced and unvoiced contributions to co-exist everywhere in the spectrum in the form of slowly and rapidly evolving waveforms, while the model used in multi-band excitation (MBE) speech coding [63] separates the speech spectrum in multiple frequency bands that can each be either voiced or unvoiced.

In many speech processing applications, the speech signal is modified in a manner that causes changes in the spectrum. Typical examples of such cases include the coarse quantization of the linear prediction coefficients in very low bit rate speech coding, the spectral smoothing at concatenation boundaries in concatenative text-to-speech (TTS) synthesis, and modification of the spectra in voice conversion where the aim is to convert the speech of a source speaker to sound as if uttered by a second speaker referred to as the target speaker. This kind of modifications performed on speech signals or their spectra may also cause unwanted changes in the effective degree of voicing if there is no explicit control on voicing. The changes in voicing, in turn, may degrade the perceptual quality of the processed speech.

In this section, we study the problem of unwanted changes in the degree of voicing and propose explicit voicing control to tackle it. The proposed approach for voicing control is implemented and experimented with in a slightly modified version of the voice conversion system presented earlier in subsection 3.1.2. The voicing control is found to offer more natural and stable voicing levels and a clear improvement in speech quality.

3.2.1 Unwanted Changes in Voicing

Many speech processing techniques utilize linear prediction (LP). For this reason, and to make the discussions easy to follow, it is assumed in this subsection that the spectral envelope of the vocal tract contribution is modeled using linear prediction. Moreover, we assume that the excitation is modeled using a sinusoidal model proposed e.g. in [24] and [150] and presented in subsection 3.1.1.

Changes in voicing

To illustrate the unwanted changes in voicing, let us assume that the original LP coefficients are modified from $\{a_j\}$ to $\{a'_j\}$ as a result of a given speech processing technique. The modification could happen e.g. due to very coarse quantization in a very low bit rate speech coding, due to spectral smoothing in concatenative TTS or due to a voice conversion related transformation. As a result of this modification, the spectral envelope changes accordingly. Assuming that the filter remains stable (that can be guaranteed e.g. by performing the modification in the line spectral frequency domain), the old and the new spectral envelopes can be directly computed based on the LP synthesis filter using

$$H(e^{i\omega}) = \frac{1}{1 - \sum_{j=1}^{K} a_j e^{-ij\omega}} , \tag{3.5}$$

and by using the same equation with the modified coefficients $\{a'_j\}$ to obtain $H'(e^{i\omega})$.

The effect that the spectral modification has on voicing can be studied by measuring the energies of the voiced and unvoiced contributions in the spectrum before and after the modification. The average energy of the voiced part for a single frame, denoted as E_V, can be estimated by sampling the spectrum at the frequencies of the sinusoids, ω_m, as

$$E_V = \sum_{m=1}^{M} \left(H(e^{i\omega_m})\nu_m A_m\right)^2 , \tag{3.6}$$

Similarly, the energy of the unvoiced contribution, E_U, can be computed as

$$E_U = \sum_{m=1}^{M} \left(H\left(e^{i\omega_m}\right)(1-\nu_m)A_m\right)^2 . \tag{3.7}$$

The corresponding energies after the spectral modifications, E'_V and E'_U, can be obtained using similar calculations as in equations (3.6) and (3.7) but by substituting $H(e^{i\omega})$ with $H'(e^{i\omega})$ to take into account the changes in the LP coefficients. It should also be noted that if the spectral modifications would cause changes in other parameters than the LP coefficients, these changes should also be taken into account when computing E'_V and E'_U.

It is usual in speech processing systems to carefully control the behavior of the overall energy but it is not common to explicitly control the relative contributions of the voiced and unvoiced components to the overall energy. However, if there is no explicit control on voicing, the spectral modifications often change the perceived level of voicing in a clearly audible way. This is caused by the fact that the relative contribution of the voiced (or the unvoiced) component to the overall energy is often changed due to the spectral modification, i.e.

$$\frac{E'_V}{E'_V + E'_U} \neq \frac{E_V}{E_V + E_U} . \tag{3.8}$$

The perceptual effect of the unwanted changes in voicing can in practice be observed as audible changes in the amplitude of the spectrally shaped noise generated to model the noise-like unvoiced contribution. This effect is discussed more closely and demonstrated using a practical example and experimental results in subsection 3.2.3.

3.2.2 Voicing Control

The unwanted changes in voicing can be corrected by controlling the voicing in an explicit way. The detailed implementation of the voicing control depends on the speech model used in the target application. In addition, even with a fixed speech model, there are different alternatives regarding the implementation.

In the case of the speech model discussed in subsection 3.1.1 and used here, one possible solution is to establish a frequency-dependent function for modifying the degree of voicing for the different sinusoids, for example as

$$\nu'_m = f(\nu_m, \omega_m, E_V, E'_V) . \tag{3.9}$$

This function can be designed in many ways and the parameters used in defining the modified degree of voicing, ν'_m, could also be different than the ones given in equation (3.9). Nevertheless, the aim is to modify the voicing of the sinusoids in such a manner that if computations similar to equations (3.6) and (3.7) would be applied again for calculating the energies after the voicing control, $\hat{E}'_V$ and $\hat{E}'_U$, we would now have

$$\frac{\hat{E}'_V}{\hat{E}'_V + \hat{E}'_U} = \frac{E_V}{E_V + E_U} \ . \tag{3.10}$$

Alternatively, it may be desired to only go towards this goal without fully satisfying it, or the target level of voicing might be decided using other techniques. E.g. in voice conversion, there could be a separate conversion model for finding out the target levels for the relative contributions of the voiced and unvoiced components based on the non-converted voicing values and possibly some other parameters, with the aim of modeling the speaker-dependencies in voicing.

The frequency-dependent operation in equation (3.9) can be used, for example, for focusing the increase in voicing more to low frequencies and/or the decrease in voicing more to high frequencies, which may be perceptually justified. However, a much simpler but still effective solution can be obtained by treating all sinusoids in the same manner. It is easy to see that the objective can be approximately achieved e.g. using the following simplified function,

$$\nu'_m = f_S(\nu_m, E_V, E'_V) = \min\left(\nu_m \sqrt{\frac{E_V}{E'_V}}, 1\right). \tag{3.11}$$

In cases where $E'_V = 0$, the voicing can be left unmodified.

Assuming that there is also a mechanism for ensuring that the overall energy stays unchanged, and that the voicing values ν_m are continuous values in the range from 0 to 1, the simplified solution presented in equation (3.11) effectively controls the level of voicing. If the voicing decisions are hard as e.g. in the MBE model, i.e. ν_m is always either 0 or 1, the best solution would be to change the voicing values of some sinusoids to approximately satisfy the condition in equation (3.10). A similar approach but with continuous voicing values could be used to complement the simplified solution in equation (3.11) to fully satisfy equation (3.10). Another solution could be obtained by also modifying the amplitudes of the sinusoids in addition or instead of modifying the degree of voicing of the sinusoids.

3.2.3 A Voice Conversion Example

In this subsection, we demonstrate the usefulness of the proposed voicing control in voice conversion. The voice conversion system used in these experiments is similar to the one described in subsection 3.1.2. The speech parameters are converted using the joint GMM approach [124] described in subsection 2.4.2 together with the data clustering and mode selection technique [44] presented briefly in subsection 3.1.2. The exception to this is the pitch parameter that is converted using the prosody conversion technique introduced in [154]. The voicing values are kept unchanged in the conversion since we have not found clear speaker-dependencies in the voicing values. The changes in the pitch are taken into account by modifying the total number of the sinusoids M and the frequencies ω_m accordingly, and by re-sampling the underlying spectral information at these new frequencies using interpolation. The system description is omitted here and the discussion is focused on the practical experiments related to the voicing control.

Effects of voicing control

In voice conversion, the spectral changes are coming from multiple sources because many, if not all, parameters are converted. However, the voicing control can still be implemented as proposed earlier in this section, provided that the changes in all parameters are taken into account when

applying the equations proposed in subsection 3.2.1 and subsection 3.2.2. The voicing control can operate directly on the voicing values without changing any other parameter values.

A practical example case demonstrating the need for controlling the voicing is given in Figure 3.1. The figure depicts the overall level of voicing, calculated as the ratio between energy of the voiced contribution and the total energy, $E_V/(E_V + E_U)$, for each 10-ms frame of an example sentence before and after voice conversion. As can be seen from the figure, the effective voicing level has clearly changed in the conversion even though the parameter values related to the degree of voicing have not been converted at all. The figure also shows that the effective level of voicing is often decreased in the conversion, leading to a higher contribution of the noise-like excitation that can be perceptually observed as increased noise. Moreover, since the difference in the level of voicing before and after the conversion is not constant, the increase in noise is also non-constant, leading to a pumping-like perceptual effect.

The unwanted changes in voicing were also studied using a larger test set consisting of 42 sentences. The overall level of voicing was measured before and after voice conversion for all the 10-ms frames of these sentences. In 47.9% of the frames, the voice conversion decreased the level of voicing, while an increase in the level of voicing occurred in 26.4% of the frames. In the rest of the frames, the level of voicing did not change due to the fact that the whole spectrum was considered either fully voiced or fully unvoiced both before and after conversion. The average level of voicing in the whole training set including all the frames was decreased due to the conversion by about 2.8%. These experimental results on a larger test set support the findings that were observed in the sentence illustrated in Figure 3.1: the voice conversion system on average decreases the level of voicing, leading to an increased level of noise in the output.

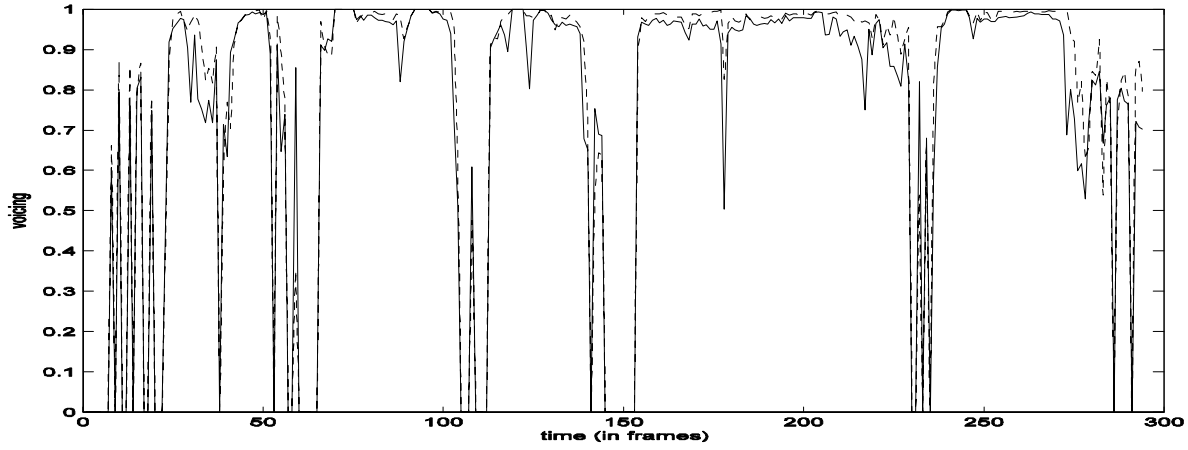

Figure 3.1: Level of voicing before (dashed line) and after conversion (solid line). (from [37])

The proposed scheme for voicing control can efficiently correct the level of voicing in such a manner that the voicing change caused by the voice conversion system is always fully compensated. The perceptual effect of having the voicing control available was informally evaluated using four expert listeners. The listeners heard converted speech samples with and without voicing control and were asked to evaluate the quality differences between the two samples. Repeated listening was possible without any restrictions. The language used in the samples was English and the voice conversion was performed between genders.

The speech quality in the samples produced using voicing control was always observed better or equal to the quality of the conventional samples. The voicing control was found to remove part of the

noise generated during the conversion. This is quite natural since the voicing control slightly reduces the contribution of the noise-like component, and it makes the behavior of that component more stable and ensures that it is better in line with the expected degree of voicing.

3.3 NOISE ROBUSTNESS TECHNIQUE

Voice conversion systems aim at converting an utterance spoken by one speaker to sound as speech uttered by a second speaker. Many different techniques have been proposed for the conversion of the speech but how to handle the pauses in the signals is a question that has not received much attention in the literature. The pauses play an important role in perception because the presence of noise is more easily perceived during the pause segments. Since speech transformation techniques tend to introduce noisy artifacts in the conversion process, the study of pause segments becomes particularly important in the context of voice conversion. Moreover, the distortion induced in the conversion process is likely to affect especially the pauses if these segments are converted with an inappropriate model trained on pure speech. This is due to the very different nature of the signals during pauses and during active speech.

It is usually beneficial to train the actual speech conversion models using only speech material representing active speech. But what should be done in the pause regions during the actual conversion process? Several straightforward techniques can be used in the conversion of the pause regions: i) to convert the pause regions using the models trained for speech, ii) not to convert the pause regions at all but to copy a segment of correct length from the source signal or from some pause signal template, or iii) to train a separate model for converting the pauses. All of these straightforward solutions have serious shortcomings. The first approach may often amplify the background noise during pauses or generate artifacts in the output. The latter two approaches rely on having a good voice activity detector (VAD) available and they are very sensitive to incorrect VAD decisions that occur even with the best VAD algorithms. Even in the case when a common model is trained for both pause and active speech, based on the existing conversion techniques, the conversion process remains highly likely to introduce distortions.

In order to improve the quality of the converted speech, general purpose speech enhancement techniques can also be considered. In general, speech enhancement algorithms first estimate noise level in the noisy speech and then eliminate it. The classical approach is spectral subtraction introduced in [155], multiple variations stemmed from it relying in fact on the same principles. The problem with such approaches is that they may not be compatible with the real-time voice conversion processing, as they are mostly used as pre-processing or post-processing techniques. Moreover, the problem addressed in this section is more specific than general speech enhancement and, as such, a tailored real-time solution is proposed that takes advantage of the particular speech parameterization.

In this section, we present a novel technique that manipulates the output in such a manner that the conversion noise that may appear during the silent regions is attenuated. At the same time, the speech quality during active speech is maintained, and the boundaries between the pauses and speech are handled in a smooth way that results in high output quality. The proposed technique has been implemented and tested in a practical voice conversion system, and it has proved to be an efficient solution to the problem. The technique has been originally published in [38].

3.3.1 Non-Speech Segment Conversion

The existing speech transformation techniques tend to introduce audible artifacts or background noise in the converted speech. The degree of perception of such distortions is closely related to the notion of signal-to-noise ratio (SNR). Assuming that the converted speech is affected by an additive stationary noise of relatively low intensity, it is intuitive that the frames with high energy would have a higher SNR than low energy frames which are perceived as more noisy. For this reason, the conversion noise can be perceived more easily during the pause segments. A possible explanation for such noise being associated with the conversion process is that usually the same conversion model is applied to both active speech and pause segments ignoring the differences between them in terms of acoustic and statistical properties. Taking these differences into account it would be sensible to treat them separately in voice conversion.

It is sensible to assume that better GMM models for voice conversion can be trained on active speech sections rather than whole utterances. If this model is to be used for the entire utterance such an approach would require special treatment in order to solve the following problems:

1. Noise artifacts may be introduced or amplified during the pauses of the converted speech.
2. Source voices (and especially TTS voices) are recorded in fairly noise-free conditions. However, the target speech is often defined by users and may be noisy in the case of mobile device recordings. This would train a noisy model which could determine additional artifacts.
3. The boundary between speech and non-speech is often ambiguous and binary VAD cannot optimally separate them in the training data. Consequently, the trained model would be negatively affected by undesired non-speech data.

3.3.2 Proposed Scheme

To overcome the drawbacks mentioned in the previous subsection, we propose a method that can train high quality models based on pure speech and can attenuate the noisy pauses in real-time enhancing the perceived speech quality and avoiding the hard decision of a VAD. This solution addresses typical voice conversion cases in which the speech signals (originals and converted) are affected by a relatively low intensity noise compared to the energy of the active speech. The goal is to enhance the quality of the converted speech starting from the observation that the noise introduced in the conversion process is particularly audible during the pause portions and that these non-speech segments have typically low energy compared to the active speech. The proposed scheme acts on the energy parameter before re-synthesis to attenuate the power of the converted signal in the low energy range. The technique converts the whole utterance, including pauses, using speech-trained models for all features (LSF, pitch, etc).

The voice conversion system used here consists of the parametric speech model described in section 3.1.1 and the joint GMM conversion technique presented in subsection 2.4.2 combined with the DTW technique for the alignment of parallel training utterances.

One of the parameters in the proposed speech representation is the energy. The energy distribution in speech and non-speech (pause) segments is depicted in Figure 3.3a. This information was collected over the full training set of the source speaker using manually supervised phonetic annotations.

Improving the quality of the conversion models

Building high quality models is possible if only pure speech frames with sufficiently large signal-to-noise ratios (SNRs) take part in the training process. The non-speech and speech with low SNR should be discarded. Assuming a stationary noise model we can infer that the energy levels of the non-speech segments give a measure of the noise level present in the signal. In order to discard the undesired frames from the training process a threshold energy can be computed and the frames with energy levels below the threshold would be discarded for being either non-speech or susceptible of low SNR. The threshold could be set, for example, just over the largest non-speech energy. It should be noted that satisfactory estimations of this value do not require the complete statistics of the entire training data but can be obtained from only a small subset if the properties of the non-speech signal are assumed to be stationary. If the speaker is a TTS voice these values can be known beforehand.

The above procedure may be applied as a pre-processing stage to improve the quality of the selected training data in cases where the original training sets are noisy. The data selected in such manner is less affected by noise and eliminates the undesired contribution of non-speech segments. This approach can be complemented by using a voice activity detector to improve the classification at low energy levels.

Reducing the conversion noise

Figure 3.3b illustrates the energy distributions after the energy feature of the source speaker has been converted using e.g. the high quality GMM determined previously, for all the utterances in the training set. The figure shows how the energies of the non-speech segments are amplified after conversion, explaining why, in some cases, the pauses are perceived noisier after conversion. This amplification is most likely caused by the conversion process influenced also by the energy distributions of the target utterances. If pause segments are converted with a model trained on pure speech, their energy is likely to increase since the pure speech has in general high energy levels. On the other hand, the increased energy levels during pauses contribute to an easier perception of the signal degradations introduced in the conversion process.

In order to attenuate the noise perceived during the pause segments we aim to reduce the value of the energy parameter in the non-speech range without affecting the intelligibility of the active speech. We note that the range of non-speech energies includes also some low-energy active speech which would be inevitably attenuated. Due to the low energy values this active speech is in a certain way masked perceptually by the high energy speech thus it has a reduced perceptual meaning and, moreover, is susceptible of low SNR. From a perceptual point of view the attenuation of these onset sections of the speech signal should not affect the intelligibility if the attenuation is done gradually.

Based on Figure 3.3b the threshold energy E_{Cmax} is empirically defined to a value larger than the highest non-speech energy. In the proposed scheme the energies below E_{Cmax} will be attenuated by a gradually increasing factor as they become closer to 0. In order to preserve the speech intelligibility, the value of E_{Cmax} should not be too high and will be determined by perceptual evaluation.

If the conversion of speech and non-speech would be done with separate models this would lead to problems related to the boundary detection and discontinuities at the transition between the two classes. On the other hand, if both classes are converted with the same model, the result is still going to be affected by conversion noise as it was shown in Figure 3.3b. The proposed conversion is accomplished with speech-to-speech models in both speech and non-speech sections, instead, the low

energy (low SNR) portions of the signal are attenuated based on an effective energy reduction technique. This is realized after the conventional conversion as a fast online extra computation that can be performed at the conversion time.

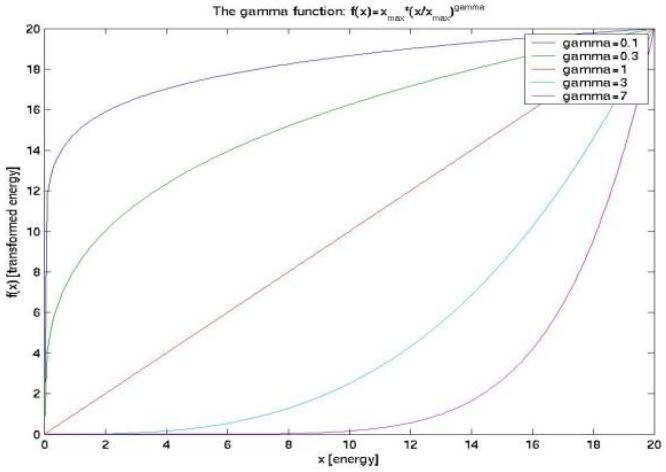

Figure 3.2: The Gamma (or Conv) function. (from [38])

The new technique applies a compression function Gamma on the converted energy feature if smaller than the threshold E_{Cmax}. The resulting energy parameter will shape the synthesis in such a way that signal power is attenuated if the converted energy is below E_{Cmax} since such frames are either non-speech or low SNR speech onset portions. The conversion of the energy parameter is realized according to:

$$Conv(E_i) = \begin{cases} \left(\frac{F(E_i)}{E_{Cmax}}\right)^{\gamma} E_{Cmax} & ,if\ E_i \leq E_{Cmax} \\ E_i & ,if\ E_i > E_{Cmax} \end{cases} \tag{3.12}$$

where $Conv(.)$ defines the proposed conversion function, $F(.)$ represents the conventional GMM-based conversion function, E_i is the energy of the current frame, E_{Cmax} is the threshold energy determined empirically, and γ is an attenuation factor.

The energy distributions after applying the proposed technique are illustrated in Figure 3.3c. We observe how the noisy pauses (non-speech) have been attenuated. The experimental results provided in the next subsection indicate that this was realized without altering the speech intelligibility and by preserving smooth transitions between speech and non-speech resulting in an overall improvement of the perceived speech quality.

3.3.3 Experiments

Figure 3.3b shows an increased energy in the non-speech regions of the converted signal as compared to the similar figure computed for the source utterances and illustrated in Figure 3.3a. This is explained by the fact that the energy distribution of the target utterances determined this amplification during the conversion process. We note that the conversion model applied to pause (non-speech) segments was trained on pure speech which has in general higher energy determining the energy amplification. Since the conversion process is also introducing noise and other signal

degradations the increased energy facilitates an easier perception of such artifacts during the pause (non-speech) portions of the signal.

Our informal listening tests led to $E_{Cmax} = 120\,dB$ and $\gamma = 2.5$. The conversion results are not too sensitive to γ in the interval $[2, 3]$ but degrade significantly if $\gamma < 2$ or $\gamma > 3$. If $\gamma < 2$ the noise is not sufficiently attenuated. $\gamma > 3$ produces too abrupt attenuation harming the intelligibility.

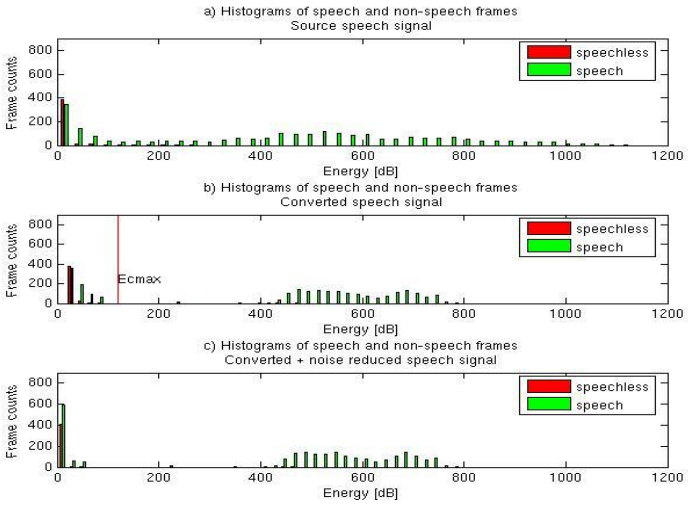

Figure 3.3: Speech vs. non-speech energy pdf: a) source b) converted c) converted + noise reduced speech. (from [38])

An informal perceptual evaluation carried out by three expert listeners indicates that the proposed method effectively attenuates the noise in the non-speech segments without negative effects on speech integrity. This is also evident in Figure 3.4 where our converted result is compared against the conventional transformation i) described in the introduction of this section. The reduction in noise levels is also illustrated in Figure 3.3c. The little amount of speech signal that was reduced in the process had already rather small energies and does not affect intelligibility or speech quality.

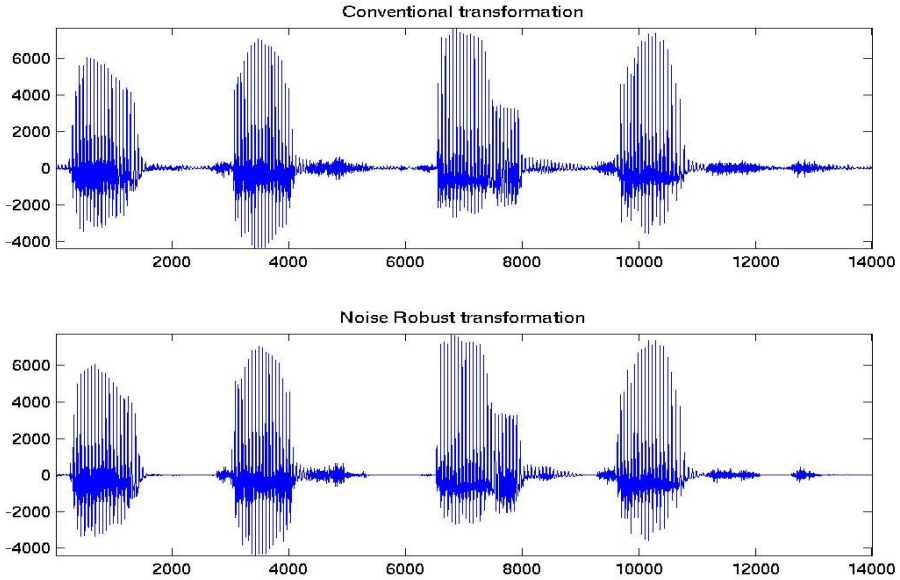

Figure 3.4: Converted waveform: a) conventional conversion b) noise reducing approach. (from [38])

3.3.4 Discussion

Adaptive extension

The technique described previously addresses the cases when the noise levels are relatively low and assumes the noise model to be stationary. If the noise or distortion present in the signal is non-stationary (remaining however of low intensity), the conversion procedure should be time-variant and use an online mechanism to update local speech and speechless models. The models of speechless and speech fragments can be iteratively updated in a local history window and thus the threshold E_{Cmax} can be updated online in an adaptive fashion. An implementation could take advantage of e.g. a voice activity detection (VAD) scheme to detect speech and speechless frames and help building the energy statistics. The accuracy of VAD decisions is not crucial since the value of E_{Cmax} is decided based on the local statistics of the energy and can always have a margin over the largest non-speech energy in the local history window.

The proposed technique has many advantages. It offers an efficient way to improve the speech quality having applicability in the selection of higher quality training data as well as during the conversion stage to enhance the perceived quality of the converted samples. The method can be implemented as a part of the conversion step serving as an automatic enhancement mechanism. Moreover, it offers a low complexity which enables real-time operation. The method avoids the storage and computation of separate models for speech and non-speech segments and all the problems associated with class detection and transitions. The technique is straightforward and flexible and can be easily adapted to different use cases. This solution takes full advantage of our parametric speech representation and can be easily integrated in the voice conversion system.

The scheme was particularly addressed to the noise levels commonly introduced by the current voice conversion techniques, aiming to enhance the speech quality in those cases. As such, the method cannot handle high levels of noise since this would imply a critical loss of speech content. Other limitations of the method include the empirical calculation of E_{Cmax}, the assumptions about noise stationarity and the sensitivity to the recording conditions. For these reasons the method works best on samples recorded in similar conditions as the training data.

3.4 CONCLUSIONS

In this chapter, a parametric speech representation was proposed as a basis for our voice conversion system. This parametric representation combines the flexibility for voice manipulation with the advantage of efficient compression. The proposed model uses line spectral frequencies to represent the vocal tract contribution and a sinusoidal model is used for the excitation signal. The suitability of this parameterization for voice conversion was evaluated using a GMM-based approach and led to promising results. Since the same parameterization is used internally by an existing TTS system, the voice conversion methods developed with this framework are directly applicable on the parametric TTS output and used to obtain customized TTS voices. Furthermore, this parametric representation is suitable to all kind of applications based on coded speech; therefore, voice conversion could be directly implemented in communications related or other embedded applications.

Since the transformations of spectral envelopes may modify also their associated voicing level, the voicing feature should be modified accordingly. Otherwise, this inconsistency will introduce an

additional quality degradation of the converted speech. A scheme for voicing control was proposed and its efficiency was demonstrated in a voice conversion system. The evaluation results indicated that this correction mechanism was able to reduce the noise level in the output signal and improved the perceptual quality.

The speech enhancement technique proposed in the latter part of this chapter provides a low-complexity, real-time solution to improving the perceived quality of the converted speech as well as a means to improve the quality of the conversion models.

As a potential future direction, we propose in the next subsection a practical solution for data collection over the phone line during normal usage. The method has several advantages such as the possibility to reach high-quality models through continuous updating since large amounts of data can be collected. Other advantages include noise robustness and easy exchange of voice models between users.

3.4.1 Future Work Proposal: A Data Collection Technique

In voice conversion, the transformation of one voice to another is made possible through a training phase that precedes the usage. Speech data from the source and target speakers are needed for the training. In general, having more good-quality data available improves the quality of the models. In practical use cases, however, it is not convenient to ask the user to provide or speak large amounts of training material. The idea of the procedure presented in this section is to make the data collection automatic and implicit in such a way that the data is collected during the normal usage of a mobile device.

An evident solution is to ask the user to speak a set of pre-defined sentences or a large amount of free speech using a dedicated tool for the recording/collection, or to provide this training corpus from the intended speaker(s), recorded in controlled conditions. If the source voice is generated using TTS, only the speech corpus from the target speaker has to be collected, but still this task can be burdensome or inconvenient for the user.

The main idea of the proposed technique is to make the collection of the voice conversion training data very easy for the user. The user has to only enable this data collection feature, and after that the system will automatically record and process data during normal usage of a mobile device without user's attention. The phone calls are recorded and further processed to obtain the training data and the voice conversion models. Additional aspects of the method enable automatic recording of other speakers and easy exchange of voice conversion models. The technique is especially designed for usage on mobile phones, and in that context, it enhances the user experience by removing the need for dedicated recordings. The technique could also be directly applied in other applications such as in internet calls and in voice messaging applications. It should be noted that the technique contains several specific aspects that require special attention, which makes normal recording applications insufficient for the task.

The proposed technique

This subsection provides only a sketch of an exemplary embodiment of the proposed technique due to its complicated nature. The details can be varied and some extra features can be added or omitted but in all embodiments the main idea is to record and process voice conversion training data during the normal usage of a device.

The implementation begins by first enabling access to the data coming from the microphone of the device (and possibly also to the data that is fed to the loudspeaker). This access has to be obtained in such a way that the normal usage of the device is not disturbed. The idea is that the method can access all the recordings done through the microphone (and possibly also all the data fed to the loudspeaker), if the user has enabled this feature, without having any effects on any aspects of the phone usage.

In addition to providing the access to the sound sources (microphone and/or loudspeaker data), the implementation of the technique needs to handle further processing of the recorded data. This step forms the heart of the technique, i.e. the automatic recording itself is not enough for realizing the technique. At least, the following aspects must be considered:

1. Inactive or too noisy parts of the data are detected and marked as non-suitable for the training corpus. The voice activity detection (VAD) can be improved in an innovative way by using dialogue modeling and by exploiting the fact that when one person is speaking during a phone conversation, the speaker at the other end is most often just listening and not talking at the same time. For example, when speakers X and Y are talking on the phone, the dialogue has four cases: 1) X speaking, 2) Y speaking, 3) both speaking, and 4) both silent. A conventional VAD for a single channel x,

$$VAD(x) = \begin{cases} 0, & x\ is\ silent \\ 1, & x\ is\ speech \\ 0 < \alpha < 1, & otherwise,\ \ soft\ decision \end{cases} \tag{3.13}$$

can be enhanced using both channels and dialogue modeling. The enhanced VAD can be simplified as

$$\begin{aligned} VAD(x) &\leftarrow [1 - VAD(y)] \cdot VAD(x) \\ VAD(y) &\leftarrow [1 - VAD(x)] \cdot VAD(y) \end{aligned} \tag{3.14}$$

where the channels x and y are recorded from speakers X and Y, respectively, and the backward arrow symbolizes the fact that the value of the right hand side term is attributed to the left hand side variable. The thresholds for rejecting recorded data through the VAD can be set in a strict manner to enhance the data quality. This simple implementation of the enhanced VAD is depicted in Figure 3.5.

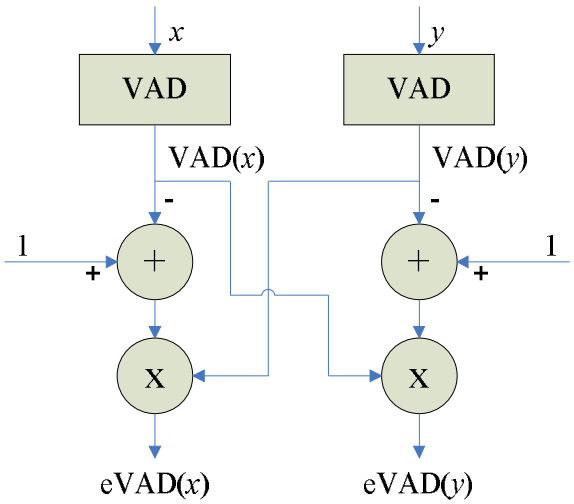

Figure 3.5: Exemplary simple implementation of the enhanced VAD that forms a part of the technique. (from [39])

2. Non-speech parts (e.g. laughter, heavy breathing) can be detected and discarded or labeled as non-speech. Again, the threshold for rejection can be set in a strict manner because the total amount of data will anyway be large when using the technique, making unnecessary rejection not harmful and potentially beneficial from the viewpoint of data quality.

3. Noise removal (and possibly some other speech enhancement) is performed on the signals that were accepted for further consideration/processing after the above rejections. Information from the enhanced VAD can be used for estimating the noise levels and statistics during silent regions, and the noise removal algorithms can be made adaptive in such a way that previously processed data is utilized in the estimation of noise statistics. The processing can also include level normalization.

4. Some light speaker verification could be applied to verify (at least approximately) that the intended speaker is talking, to ensure that the training sets remain speaker-dependent. The reference data for the speaker verification could be collected during the first recorded call or as a separate step when the user enables the data collection. Again, the threshold for accepting the speaker verification result can be set in a strict manner to ensure that the data is useful and from the intended speaker with a very high probability. Since communication devices such as mobile phones are usually dedicated for personal usage by the owner, it is easy to obtain good performance in the speaker identification even with simple implementation, as it can be considered highly likely that the same person is speaking every time.

5. The speech is analyzed and represented in the domain in which the voice conversion will be performed. The analysis and the representation follow the framework presented earlier in section 3.1.1. The resulting data is included in the training corpus.

6. The conversion models are trained using the data in the correct domain. Any prior art or new techniques can used in the training, and there are no limitations on the types of models used in the conversion scheme.

7. To save memory, the training data itself does not have to be stored. Instead, the trained model can be stored. When more data becomes available, e.g. after another phone call, the model can be further updated/improved based on the new training data. It is also possible to delay the model training until some pre-specified amount of data has been collected. If this limit is reached during a phone call, the data collection can either be continued or stopped. In the former case, the data size can be allowed to grow or it is also possible to implement a mechanism that aims at replacing older data of worse quality with better-quality new data after the limit is reached. E.g. there can be two limits for the size of the data: the lower limit is set to ensure sufficient amount of data for reliable training and the higher limit for ensuring that the data size does not grow too much. Whenever the size of the stored data (plus the size of the new data to be considered) is between the limit, the data with the lowest estimated quality can be discarded until the lower limit is reached. An exemplary implementation of this kind of a memory saving scheme is presented in Figure 3.6a.

8. The trained models can be used in voice conversion. If the user finds the model good enough, she/he can disable the data collection functionality.

The whole process forming the technique is also illustrated in Figure 3.6b.

It may often be the case that the user does not want to hear her/his own voice but the voice of someone else (e.g. the voice of a spouse). To make this possible, the core method can be complemented with a feature that allows users to share their voice data/models. Different users can collect their own voice data using the proposed method and then share the trained models with each other. The sharing can be done by transmitting the model parameters between two devices via some

means of connectivity. It is possible to implement this feature of the technique in such a way that everyone can freely distribute only their own models and a specific permission is needed for creating freely-distributable models.

An extra feature of the technique could be the possibility to record also incoming speech data (the data that is fed to the loudspeaker during a phone call) for some particular speakers (controlled by the user), i.e. to record speech from the other end during phone calls. In this case, it should be noted that the speech has already been compressed and decompressed for the transmission but the most critical features of speech should be quite well preserved. The identity of the speaker can to some extent be guessed directly based on the phone number but the user could also be given a chance to reject the content of some call from the training e.g. in a case when someone else was using the phone of the intended speaker. It is also possible to design a speaker verification scheme that exploits the information that it is highly probable that the owner of the phone is the speaker. Despite the technical feasibility, however, it is not 100% evident whether this kind of recording of the received voice data without the consent of the other speaker could result in legal problems in some countries or in some circumstances. If there is any doubt, this technique can be implemented in such a way that by default the signal from the other end is not recorded, and some form of permission is required before starting the recording. In any case, the data from the other end can be used for enhancing the VAD performance as discussed in item 1 above.

Discussion

The proposed technique has clear advantages in terms of user experience and quality of trained models. Enhanced user experience is provided through very easy data collection, means of sharing voices among friends etc. The technique offers the possibility to generate high quality models based on large amounts of data and to improve existing models through the collection of additional training data.

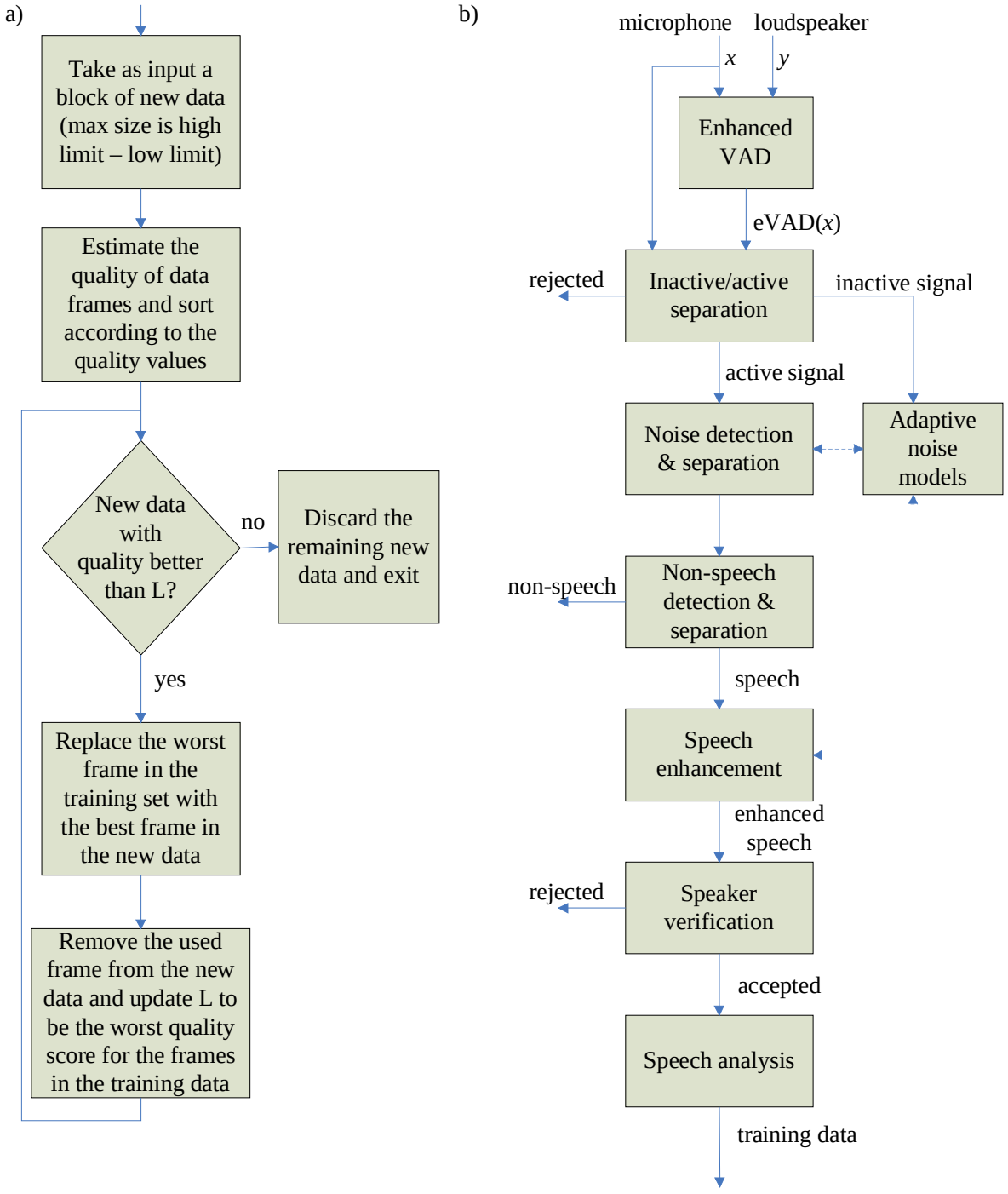

Figure 3.6: a) Exemplary implementation of the memory saving scheme needed in the implementation of the technique. L denotes the estimated quality value for the worst data frame in the training data. b) Overview of the technique. (from [39])

Chapter 4

Proposed Alignment Techniques

In a standalone voice conversion system a transformation function is learned typically from a set of paired vectors of the source and target speakers. The process which realizes this pairing is called alignment and represents an important part of a voice conversion system. The alignment plays an important role in the versatility of the system and different strategies are adopted depending on the type of training corpus. In the simplest case, when the source and target speakers utter the same sentences, the corpus is referred to as parallel or text dependent and dynamic time warping (DTW) has been widely adopted as the alignment technique. In this chapter a soft alignment technique [41] is proposed as an alternative to DTW promising to overcome some limitations inherent to the binary nature of DTW.

In real situations the parallelism is a restrictive requirement for the training data and non-parallel (or text independent) data is more easily available. In spite of that, non-parallel data poses greater challenges for alignment since the speakers do not utter the same utterances. In the most extreme case of text independent voice conversion the two speakers speak different languages that may have different phoneme sets. Theoretically, text independent voice conversion can be reduced to the parallel scenario by a proper alignment and similar conversion methods can be applied. Due to alignment limitations the performance levels in these cases have generally been lower. In this chapter two techniques for text independent alignment are proposed: one is based on temporal decomposition [40] and the second uses an intermediary stage based on text-to-speech synthesis [42].

The chapter is organized as follows. In section 4.1 the soft alignment scheme is introduced. Section 4.2 presents the text independent alignment based on temporal decomposition and in section 4.3 the main conclusions are summarized and a TTS-based technique for text independent alignment is described as a future direction.

4.1 SOFT ALIGNMENT SCHEME

The conventional GMM based approach can be used successfully in feature transformation for example in voice conversion. However, in GMM based transformation, the alignment between source and target vectors is a very crucial factor affecting the quality since aligned vectors are needed

for GMM training. For example, in the voice conversion application, equivalent utterances from source and target speaker are used as the training material. The fine-scale alignment can be either done manually or automatically by dynamic time warping (DTW). Though those alignment methods have been commonly used, they have the following drawbacks due to their inherent hard alignment strategy.

1. Given any source-target pair of vectors, there are only two possible decisions in the alignment: the pair is assumed 100% aligned or it is not aligned at all. As a result, even small alignment errors can introduce a lot of noise to the data to be modeled.
2. Hard alignment may be impossible by nature. For example vectors extracted from human speech cannot always be perfectly aligned even by human experts using the conventional hard alignment due to the properties of natural speech. Thus, the hard alignment will always cause some errors no matter how well it is performed.
3. Hard alignment requires that in order to obtain the final vector sequences, the aligned sequences from the source and the target must have the same number of vectors. Therefore, it is necessary to interpolate between vectors or to discard/duplicate some vectors. Both of these solutions add more complexity and introduce alignment or interpolation errors.

The quality of GMM is greatly affected by "noisy" data due to the alignment problems. The work on alignment has typically concentrated on improving hard alignment manually by experts or through the use of different automatic techniques, such as dynamic time warping. The proposed technique can bring improvements by introducing a soft alignment method.

The main idea of the technique is to utilize soft alignment to obtain the training data for GMM based transformation. In soft alignment, alignment probabilities are assigned to vector pairs. The soft alignment probabilities are then used in the training. A very good use case for the technique is voice conversion (the transformation of the characteristics of a source speech to match the characteristics of a target speech).

4.1.1 The Proposed Method

The proposed soft alignment can be realized in many ways. In this section, we introduce one possible reference realization designed for the voice conversion application but it should be noted that this realization is only exemplary. In all the alternatives, the core of the technique is that vectors are aligned using soft alignment, i.e. a feature vector may align to other feature vectors with some alignment probability. GMM is then trained using joint feature vectors and the corresponding alignment probabilities.

Soft alignment

In the present technique, an alignment probability $PA(x_p, y_q)$ is introduced to express that a feature vector x_p at time p from the source speaker is aligned to the feature vector y_q of the target speaker at time q with probability $PA(x_p, y_q)$, to form the joint feature vector of the aligned pair $z_k = [x_p{}^T \; y_p{}^T \; PA(x_p, y_q)]^T$. For hard alignment, $PA(x_p, y_q)$ would be 1 for all aligned pairs. For soft alignment, $PA(x_p, y_q)$ is continuous value between 0 and 1. In the following two subsections, we propose a novel algorithm for soft alignment and GMM training as a reference realization of the proposed scheme. It can be implemented in several other ways as well. The key is taking soft alignment into account in some way.

Alignment probability estimation

In Hidden Markov Model (HMM) based speech recognition, it is assumed that the sequence of speech feature vectors corresponding to each word is generated by a Markov Model. A Markov Model is a left-to-right finite state machine which changes state once every time unit. For a sentence, HMMs are concatenated together to have bigger HMM consisting of left-to-right state in a sequence.

The same ideas can be used for deriving alignment probabilities for the voice conversion application. Let a given source and target speech be represented by a sequence of feature vectors, $\boldsymbol{x} = [x_1\ x_2 \dots x_t \dots x_n]$ and $\boldsymbol{y} = [y_1\ y_2 \dots y_t \dots y_m]$, respectively, where x_t, y_t are speech vectors (e.g. Mel-scaled frequency cepstral coefficients, the so-called MFCC features) at time t. A compound HMM model is generated by concatenating all subword HMM models (e.g. speaker-independent phoneme based HMMs for intra-lingual language voice conversion, or even language-independent phoneme based HMMs for cross-lingual voice conversion task. The units smaller than a phoneme could also be possible to capture more precise time information) given a pronunciation transcription of the source and the target speech. Typically a phoneme HMM composed of several states is used. The joint probability that a sequence $\boldsymbol{x}$ is generated by the compound HMM model M through the state sequence $x(t)$ can be simply calculated. $x(t)$ denotes the state index that feature vector x_t at time t is aligned to. In practice, the underlying state sequence is, however, unknown (or hidden). Let $LS_j(x_t)$ and $LT_j(y_t)$ denote the probability of the j^{th} state occupation at time t for the source and the target. This can be efficiently calculated using the so-called forward-backward algorithm. Suppose we have the compound HMM model M for a given sequence $\boldsymbol{x}$ of feature vectors from the source speaker. Let the forward probability $\alpha_j(t)$ and the backward probability $\beta_j(t)$ for model M with N states be defined iteratively as:

$$\alpha_j(t) = P(x_1, \dots, x_t, x(t) = j|M) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1) \cdot a_{ij}\right] \cdot b_j(x_t) \tag{4.1}$$

$$\beta_j(t) = P(x_{t+1}, \dots, x_n|x(t) = j, M) = \sum_{i=2}^{N-1} a_{ji} \cdot b_i(x_{t+1}) \cdot \beta_i(t+1) \tag{4.2}$$

Where $P(x_1, \dots, x_t, x(t) = j|M)$ denotes the probability for observing the sequence $x_1, \dots, x_t$ with the speech vector x_t at time t corresponding to state j $(x(t) = j)$ given the model M. Thus, x_t would be generated from the probability density b_j of state j. The transition from state i to state j is also probabilistic and is governed by the probability a_{ij}. $P(x_{t+1}, \dots, x_n|x(t) = j, M)$ is the probability for observing the sequence $x_{t+1}, \dots, x_n$ given the model M and the fact that x_t was generated from state j. Hence,

$$LS_j(x_t) = \frac{\alpha_j(t) \cdot \beta_j(t)}{P(\boldsymbol{x}|M)} = \frac{\alpha_j(t) \cdot \beta_j(t)}{\sum_{j=2}^{N-1} \alpha_j(t) \cdot \beta_j(t)} \tag{4.3}$$

where $P(\boldsymbol{x}|M)$ denotes the probability to observe the sequence $\boldsymbol{x}$ given the model M.

The probabilities for state occupations are computed for both the source and the target speech. Figure 4.1 illustrates this process.

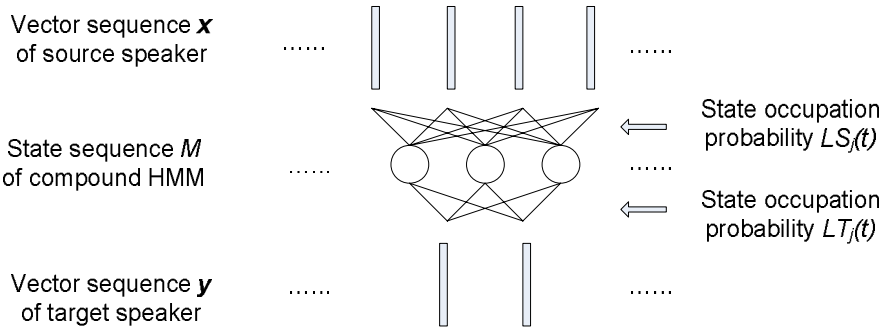

Figure 4.1: State occupation probabilities for the source and the target speech. (from [41])

As introduced above, the state occupation probabilities can be calculated using the forward-backward algorithm. For soft alignment, the alignment probability $PA(x_p, y_q)$ has to be calculated to measure the probability that source vector x_p can be aligned to target vector y_q. Now we have,

$$\begin{aligned} PA(x_p, y_q) &= \sum_{l=1}^{L} PA(x_p, y_q | x(p) = l, y(q) = l) \\ &= \sum_{l=1}^{L} \Big(PA(x_p | x(p) = l) \cdot PA(y_q | y(q) = l) \Big) = \sum_{l=1}^{L} LS_l(x_p) \cdot LT_l(y_q) \end{aligned} \tag{4.4}$$

where L is the number of HMM states in a given sentence and e.g. $PA(x_p|x(p) = l)$ denotes the probability of observing the p^{th} source vector x_p being generated from (or aligned to) state l.

In this implementation of the soft alignment technique, the alignment probability $PA(x_p, y_q)$ is calculated using equation (4.4).

GMM estimation

The alignment probability is introduced to express that a feature vector x_p at time p from the source speaker is aligned to a feature vector y_q at time q of the target speaker with a soft probability $PA_{pq} = PA(x_p, y_q)$, so we have joint vectors defined as $z_k = z_{pq} = [{x_p}^T \; {y_p}^T \; PA_{pq}]^T$. It should be noticed that feature vectors for soft alignment and features for GMM training and conversion can be different. Once the alignment probabilities are obtained, they can then applied to features commonly used for voice conversion, such as line spectral frequency (LSF), etc. The parameters of GMM are estimated using the Expectation Maximization (EM) algorithm. In the following, we give a detailed description of an estimation procedure that uses the proposed soft alignment technique.

Expectation step:

For mixture component number $l = 1, \dots, L$, the posterior probability, given observation z_{pq}, is:

$$\begin{aligned} P_{l,pq} &= P(l | z_{pq}) = \frac{P(z_{pq}|l) \cdot P(l)}{P(z_{pq})} \\ P(z_{pq}) &= \sum_{l=1}^{L} P(z_{pq}|l) \cdot P(l) \\ \tilde{P}_{l,pq} &= PA(x_p, y_q) \cdot P_{l,pq} \end{aligned} \tag{4.5}$$

where $P_{l,pq} = P(l|z_{pq})$ denotes the posterior probability of component l given the observation z_{pq}, $P(z_{pq}|l)$ is the likelihood of z_{pq} given the pdf of the l^{th} Gaussian component, $P(l)$ is the prior

probability of the l^{th} Gaussian component. $\tilde{P}_{l,pq}$ is the updated version of $P_{l,pq}$ based on the current iteration step.

Maximization step:

$$\begin{aligned} \tilde{P}(l) &= \frac{1}{m \cdot n} \cdot \sum_{p=1}^{n} \sum_{q=1}^{m} \tilde{P}_{l,pq} \\ \tilde{\mu}_l &= \frac{\sum_{p=1}^{n} \sum_{q=1}^{m} \tilde{P}_{l,pq} \cdot z_{pq}}{\sum_{p=1}^{n} \sum_{q=1}^{m} \tilde{P}_{l,pq}} \\ \tilde{\Sigma}_l &= \frac{\sum_{p=1}^{n} \sum_{q=1}^{m} \tilde{P}_{l,pq} \cdot (z_{pq} - \tilde{\mu}_l) \cdot (z_{pq} - \tilde{\mu}_l)^T}{\sum_{p=1}^{n} \sum_{q=1}^{m} \tilde{P}_{l,pq}} \end{aligned} \tag{4.6}$$

where m and n represent the lengths of the source and target vector sequences. $\tilde{P}_l$, $\tilde{\mu}_l$ and $\tilde{\Sigma}_l$ are the prior probability, the mean and the covariance matrix of the l^{th} Gaussian component, respectively.

The conversion function that converts source feature x to target feature y is given by:

$$F(x) = E[y|x] = \sum_{l=1}^{L} p_l(x) \cdot \left(\tilde{\mu}_l^y + \tilde{\Sigma}_l^{yx} \left(\tilde{\Sigma}_l^{xx} \right)^{-1} (x - \tilde{\mu}_l^x) \right) \tag{4.7}$$

where $E[.]$ denotes expectation and $p_l(x)$ is the posterior probability of the l^{th} component given the observation x.

4.1.2 Experiments

Some preliminary basic tests were carried out in a voice conversion framework to prove the advantages of soft alignment over the hard alignment.

Voice conversion generally requires the alignment of parallel data as a preprocessing stage. A mapping model is trained on the aligned data and further used in conversion. When parallel test data is available, the converted features can be benchmarked/evaluated against aligned target features in terms of mean squared error (MSE). In this experiment we used 16 kHz English language speech samples taken from a TC-Star parallel corpus [151].

Pitch and LSF features were transformed in our experiments as they retain the essential identity content of the speech. An initial dataset is assumed to have ideally aligned features, which will be regarded as generic data ignoring their speech related meanings. Artificial data is derived by a *decimation + pairing* procedure from the initial set. The *decimation + pairing* procedure is distinct for soft and hard alignments. Two distinct artificial sets will thus train distinct models for the hard and soft alignments. The source data in the initial set is transformed through both soft and hard models and compared to the (initial) target in terms of MSE.

In this experiment, the source and the target were identical, except for decimation and pairing. Note that in this case the alignment is truly ideal. In creating the artificial data the *decimation step* is identical for hard and soft cases: decimation by 2 is applied to distinct parities for source and target data in the initial set (see Figure 4.2 and Figure 4.3). The *pairing*, however, is different: hard alignment on the decimated data is nothing but a one-to-one mapping between the corresponding indices whereas in soft alignment each sample in the target is paired with equal probabilities (0.5) to its closest two feature vectors in the source samples (see Figure 4.2 and Figure 4.3).

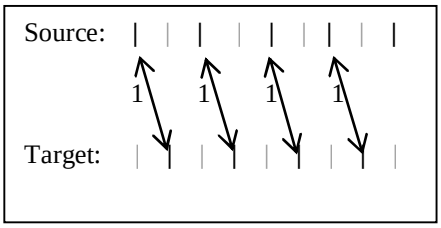

Figure 4.2: Hard alignment. (from [41])

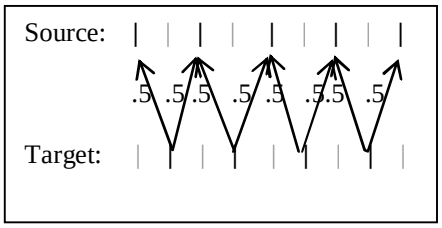

Figure 4.3: Soft alignment. (from [41])

For pitch the *decimation + pairing* procedure needs to be done for each voiced segment individually to avoid cross-segment pairings. Yet for LSF the procedure is applied only once for the entire dataset (the initial set) since LSF are continuous over voiced and unvoiced regions in our representation.

The MSE computed for LSF is based on the definition given in 6.3.3 averaged over the test set. A similar figure is computed for the pitch exclusively over the voiced frames, where pitch is regarded as a vector with one element. The MSE results are given in Table 4.1.

Table 4.1: Hard vs soft alignment: GMM performance measured using MSE

	Hard Alignment	Soft Alignment
Pitch (voiced)	1.42	0.26
LSF (all)	749.1	479.8

The superiority of the proposed soft alignment is outstanding when the probabilities are correctly assigned. The experimental result achieved with the simplified probability assignment (up to 82% MSE reduction) was quite impressive considering that the size of the context was only two feature vectors in terms of alignment probability. The results show that soft alignment has significantly outperformed the classical hard alignment.

4.1.3 Discussion

The proposed soft alignment technique does not require making any strict assumptions about data and has numerous advantages such as the reduction of the alignment errors and the increased robustness of the resulting GMM. The technique produces more combined feature vectors than hard alignment therefore more data will be available for training. In soft alignment there is no need for interpolation or for removing or duplicating vectors. On the downside the computational complexity is arguably higher than with the prior art alignment although this is not a critical drawback. No other significant disadvantages of the proposed idea could be identified.

4.2 TEXT INDEPENDENT ALIGNMENT BASED ON TEMPORAL DECOMPOSITION

In contrast to the parallel scenario, in text independent corpora the speakers need not utter the same sentences. If the speakers use the same phonetic set in their utterances, we refer to this as the non-parallel case. In a cross-lingual case even the phoneme sets used by speakers are different. In our experiments, we have designed a so called simulated cross-lingual corpus from originally parallel intra-lingual data by ensuring that the target speaker utilizes in the utterances only a subset of the phonemes used by the source speakers.

By observing that the correspondence of speech content between speakers has moved from the utterance level to the phoneme level, we propose an alignment scheme based on temporal decomposition and phonetic segmentation of the speech signal inspired by a similar alignment approach introduced in [16]. The role of the alignment scheme in this work is to facilitate the use of parallel voice conversion algorithms with text independent data allowing us to focus on the evaluation of GMM and the bilinear models introduced later in section 6.3. Therefore, we limit ourselves to presenting the scheme and leave further analysis for future study.

4.2.1 Temporal Decomposition

In the temporal decomposition (TD) model [156], speech is represented as a sequence of articulatory gestures that produce acoustic events. An acoustic event is associated with a so called event target and with an event function. The event target can be regarded as a spectral parameter vector and the event function denotes the activation level of that acoustic event as a function of time. The mathematical formulation of this model was given in [156] as:

$$\hat{y}(n) = \sum_{l=1}^{L} z_l \phi_l(n), \qquad 1 \le n \le N, \tag{4.8}$$

where z_l denotes the l-th event target, $\phi_l(n)$ describes the temporal evolution of this target, $\hat{y}(n)$ is an approximation of the n^{th} spectral parameter vector $y(n)$, N is the number of frames in the speech segment and L represents the number of event functions, $(N \gg L)$. Let P denote the dimension of the vectors $y(n)$ and z_l.

In the original formulation of the TD model [156] several event functions may overlap at any given location in the speech signal. A simplification of the original model was proposed in [157][158] in which only adjacent event functions are allowed to overlap leading to a second order TD model:

$$\hat{y}(n) = z_l \phi_l(n) + z_{l+1} \phi_{l+1}(n), \qquad n_l \le n \le n_{l+1}, \tag{4.9}$$

where n_l and n_{l+1} represent locations of events l and $(l + 1)$, respectively.

A restriction on this model suggested in [159] requires the event functions to sum up to one. Furthermore, in order to better explain the temporal structure of speech, a modified restricted temporal decomposition (MRTD) was proposed in [160] and assumes that all event functions first grow from 0 to 1 and then decrease from 1 to 0. An illustration of the event functions with all the above restrictions is given in Figure 4.4.

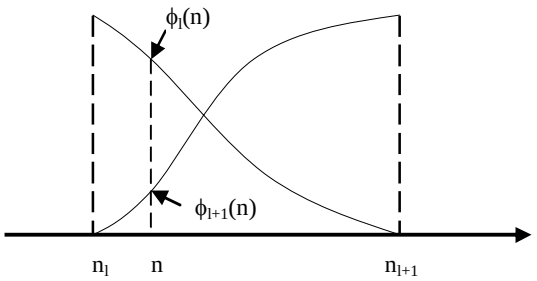

Figure 4.4: Two adjacent event functions in the second order TD model. (from [40])

The above assumptions are practically equivalent to saying that any spectral vector located between two event targets can be computed from the event targets by interpolation.

The MRTD algorithm [160] can be used to determine the event locations and the event targets. Interestingly, [161] suggests that these event targets convey speaker identity. However, MRTD cannot guarantee a fixed correspondence between the (number of) acoustic events and the phonetic units. Such a property is desirable for alignment purposes [16] but requires another method to find the event locations.

A method based on phonemes was proposed in [162][163] to represent phonemes with a fixed number of event targets. The method uses labeled utterances to segment the speech signal into phonemes. Each phoneme is divided into $(Q-1)$ equal segments by Q equally spaced points which are used as event locations. Q is a free parameter depending on the application. In [16] $Q = 5$ was used.

In our work we distinguish between the middle stationary part of phonemes and phonetic transitions and segment the speech into stationary phonetic units and transient phonetic units as described in the next section. Each phonetic unit is divided by four equally spaced points ($Q_{pu} = 4$), corresponding to seven event targets per phoneme ($Q = 7$), and event targets are computed at those locations from an LSF representation of the phonetic unit as follows [160].

First the event functions are computed as:

$$\phi_l(n) = \begin{cases} 1 - \phi_{l-1}(n), & if\ n_{l-1} < n < n_l \\ 1, & if\ n = n_l \\ min\left(\phi_l(n-1), max\left(0, \hat{\phi}_l(n)\right)\right), & if\ n_l < n < n_{l+1} \\ 0, & otherwise \end{cases}, \tag{4.10}$$

where $l = 1{:}\ Q_{pu}$ and

$$\hat{\phi}_l(n) = \frac{\langle (y(n) - z_{l+1}), (z_l - z_{l+1}) \rangle}{\|z_l - z_{l+1}\|^2}, \tag{4.11}$$

in which $\langle .,. \rangle$ and $\|.\|$ denote the inner product of two vectors and the vector norm, respectively, $y(n)$ represents the n^{th} vector of the LSF sequence and the initial event targets z_l and z_{l+1} are vectors sampled from the LSF trajectories at the defined target locations n_l and n_{l+1}, $z_l = y(n_l)$, $z_{l+1} = y(n_{l+1})$.

The actual event target vectors are then calculated in least mean square sense using:

$$Z = Y\Phi^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}})^{-1}, \tag{4.12}$$

where the matrices $Y \in \mathbb{R}^{PxN}$ and $Z \in \mathbb{R}^{PxL}$ contain the spectral parameter vectors and event vectors, respectively, one per column, and $\Phi \in \mathbb{R}^{LxN}$ consists of the event functions arranged line-wise; with L, P and N defined in the beginning of the subsection. Since these event target vectors may violate the frequency ordering property of LSFs a further refinement scheme is applied as in [160].

The event targets are used for alignment and in conversion while the reconstruction of the phonetic unit to its original or to a desired number of feature vectors is done based on the event functions.

4.2.2 The Proposed Alignment Scheme

The next scheme requires phonetically labeled training data and aims to align a cross-lingual training corpus with two source speakers covering the same phoneme set and one target speaker whose phoneme set is different. The concept of voice conversion with multiple source speakers will be discussed in more detail in section 6.3.

The idea is to segment the training speech into central (assumed stationary) parts of the phonemes and transitional parts and organize this data into a multi-layer matrix in which each layer corresponds to a different speaker, diagonal nodes correspond to stationary parts and all other nodes correspond to transitional parts. Once the data is organized in this way, for any given node we apply some similarity criterion to retain in every layer an equal number of signal/parameter instances in order to achieve a one-to-one frame correspondence between all the speakers.

Let $\Theta = \{\theta_1, .., \theta_P\}$ denote a phonetic set consisting of phonemes *common* to all source and target speakers and *rare* phonemes spoken only by the source speakers. The notation θ_f for a phoneme in this set is used to indicate its order index f within the set. If $\theta_f \in \Theta$ is the j^{th} phoneme in a phonetic transcription of an utterance we will refer to θ_f also as p_j in order to describe its position in the phonetic sequence.

If $\theta_f\ (= p_j)$ occupies the time interval $[\tau_{j-1}\ \tau_j]$ in the speech signal of a given utterance, we define a stationary phonetic unit $p_j p_j (= \theta_f \theta_f)$ to be signal portion situated in the time interval $[\tau_{1,j}\ \tau_{2,j}]$ where:

$$\tau_{1,j} = \tau_{j-1} + 0.25 \cdot (\tau_j - \tau_{j-1}) \quad \text{and} \tag{4.13}$$

$$\tau_{2,j} = \tau_{j-1} + 0.75 \cdot (\tau_j - \tau_{j-1})\ . \tag{4.14}$$

In other words, the stationary phonetic unit is by definition the central part of a given phoneme occupying the second and third quarters of the phoneme signal. This extraction of the spectrally stable parts is best justified for vowels but it is used here for the rest of the phonemes as well.

The transient phonetic unit $p_{j-1}p_j (= \theta_g \theta_f)$ occupies the interval $[\tau_{2,(j-1)}\ \tau_{1,j}]$ instead, consisting of the signal portion left between the stationary units $p_{j-1}p_{j-1}$ and $p_j p_j$.

Following the procedure in the previous subsection, the (LSFs of) phonetic units are decomposed into $Q_{pu} = 4$ equally spaced event targets which can be concatenated into a phonetic unit based feature vector

$$Z = [z_1^T\ z_2^T \ldots z_{Q_{pu}}^T], \tag{4.15}$$

where $z_q\ (1 \le q \le Q_{pu})$ denotes the q^{th} event target in a speech segment (phonetic unit). If we consider the frequency ordering property of the "LSF-like" event targets, this representation can be further normalized to

$$Z = [z_1^T\ \ z_2^T + \pi\ \ \ldots\ \ z_{Q_{pu}}^T + (Q_{pu} - 1)\pi], \tag{4.16}$$

which is an ordered vector of frequencies within the interval $(0\ Q_{pu}\pi)$.

All the phonetic unit based feature vectors are then grouped by phonetic unit and speaker. We represent our training data in the form of a PxP matrix D, structured in multiple layers (one for each speaker) having at node (g, f):

- the stationary phonetic unit $\theta_f \theta_f$ corresponding to phoneme θ_f, if $g = f$
- the transient phonetic unit $\theta_g \theta_f$ between phonemes θ_g and θ_f, if $g \neq f$

Aligned data is built for each phonetic unit by grouping phonetic unit based vectors from each layer of the unit's node (g, f) into triples (Z_{s_1}, Z_{s_2}, Z_t) minimizing the distance $SD_3(Z_{s_1}, Z_{s_2}, Z_t) = sd(Z_{s_1}, Z_{s_2}) + sd(Z_{s_1},\ Z_t) + sd(Z_{s_2}, Z_t)$ over all combinations of the remaining vectors at node (g, f) until one layer runs out of phonetic unit based vectors. Here $sd(.)$ represents a spectral distortion measure. Consequently, we end up with an equal number of phonetic unit based vectors in each layer. A node (g, f) in which at least one of θ_g or θ_f is a *rare* phoneme cannot contain data

from the target speaker, therefore we align phonetic unit based vectors as pairs (Z_{s_1}, Z_{s_2}) only between source speakers' layers. This is done in a similar way minimizing the distance $sd(Z_{s_1}, Z_{s_2})$ over all the remaining combinations until one layer runs out of data.

This alignment strategy has been used in a practical voice conversion application and the results are presented in detail in section 6.3. Since that study discusses the bilinear models as a central topic, the alignment scheme was not the main focus and, as such, was not evaluated against an alternative alignment scheme. However the perceptual tests which compared non-parallel and parallel voice conversion found them fairly similar. This indicates the efficiency of the proposed technique since the alignment of parallel data has become almost standard and can be considered to offer currently an upper limit for the performance.

4.3 CONCLUSIONS

While DTW has become almost a standard technique providing sufficient accuracy for the alignment of parallel data, the solution for the non-parallel case is not straightforward and attracted an increasing research interest lately. Different techniques have been proposed for both intra-lingual and cross-lingual data performing in general below the level of the parallel solutions.

In this chapter the rather minor but inherent limitations of DTW given by its hard binary nature were challenged and a soft alignment technique was proposed. The idea is to allow a source frame to map with different probabilities to several target frames instead of a hard one-to-one mapping and vice-versa. This would avoid the need for interpolation between vectors or vector duplicates as is the case with DTW and would implicitly generate more training data as a side benefit. The idea was demonstrated viable in experiments with an artificial example of parallel data.

The non-parallel case has also been addressed. The proposed technique requires phonetic segmentation and defines stationary and transient phonetic units which are then decomposed into a fixed number of event vectors and organized in a matrix structure. This alignment scheme has been used and proved functional in a practical voice conversion scheme discussed later in section 6.3 dedicated to the study of bilinear models. In that context, the alignment was not in focus and, due to this fact, the method was not evaluated in comparison with a separate alignment scheme. In comparison with an equivalent system trained from an equal amount of parallel data, the proposed technique led to perceptually similar results in a subjective listening test. This aspect indicates the efficiency of the method since the parallel scenario can be considered an upper limit of the performance.

In the next subsection we introduce and propose for future work a method that uses a TTS system to divide the non-parallel conversion problem into two sub-problems with parallel data. The parallel conversion models are then concatenated. The method relies on well studied voice conversion techniques and is very attractive for its improved usability and ease of implementation.

4.3.1 Future Work Proposal: Text Independent Alignment Using TTS

Typically, voice conversion systems are trained using aligned speech material from the source and target speakers. On the other hand, both the source and the target speaker are asked to speak the same sentences. Concerning the practical use cases and the usability of this approach in general, the requirement of having parallel training material is a very limiting and inconvenient aspect. It would

be much easier for the users and developers to obtain speech material if free speech could be used, without the need to restrict the sentences. Consequently, there is a clear need for developing techniques that would enable text-independent voice conversion on general purpose. The term text-independent refers to the fact that there is no limitation on the sentences that the speakers read/speak for the training. To be able to find the parallel subunits from both source and target sides in similar contexts, in practice the database has to be usually bigger than one required for the parallel case.

A number of prior-art approaches have been proposed in the literature for aligning the vectors of a text-independent corpus. These techniques tend to perform worse than the parallel scheme [109] or at best, in recent methods [111][112], approaching to a small difference and many of them require linguistic knowledge for tuning the system [106][111]. In [109] the source and target vectors are independently clustered and the correspondence is established, first between clusters based on the similarity of their frequency warped centroids, and then, at frame level within the clusters. This method lead to spectral distortion results between 15% and 25% worse compared to the parallel case. Speech recognition is used in [106] to label all the source and target frames with a state index and the alignment is performed by finding the longest matching sequences from the two speakers. This approach is limited to intra-lingual cases and may be affected by recognition accuracy problems. The alignment proposed in [110] uses a unit selection approach resembling a TTS for which the synthesis database consists of the target speaker's frames. The main drawback of this approach is that the output tends to be increasingly similar to the source voice for larger data. Recently, in [111] a phoneme cluster correspondence between the source and target acoustic spaces is used to initialize a self-organizing iterative algorithm which learns a topology preserving mapping that maps neighboring inputs to neighboring outputs. In [112] a data driven approach is introduced based on the iteration of some existing voice conversion techniques. The results reported with the latter two methods approach to a marginal difference to the parallel voice conversion. Another prior art method [104] limits the TTS voice as either source or target in the text independent voice conversion system, so that the solution cannot be extended to the general applications where both source and target speakers are human speakers. The technique proposed in this section is targeted for the case where both the source and the target are natural speakers. According to the author's knowledge, a more generalized text independent scheme that could use speech synthesis as intermediate step to divide the non-parallel training into two parallel steps has not been used or published anywhere.

The state-of-the-art text independent voice conversion techniques are set the objective to find frames that correspond to each other. The current technique proposes a novel approach to build up the text independent voice conversion training procedure. A high-quality speech synthesis system is used as intermediate step to bridge the non-parallel gap between utterances from the source and target speakers. On the other hand, the non-parallel text independent voice conversion from the source speaker to the target speakers can be divided into two parallel voice conversion parts. At first, the source speech is converted into the corresponding the TTS speech in parallel. Then the intermediate TTS speech is converted into the target speech in parallel too. This is based on the fact that the training procedure is done offline and the high-quality TTS system is readily available for use even on the same device. The TTS system can be used for artificially and simply generating training data for voice conversion.

Given the non-parallel training utterances from the source and target speakers, the TTS training utterances can be made available so that they are parallel to the utterances from the source speaker and the target speakers. The source-to-TTS conversion model can be trained using parallel training

corpus between the source and TTS speeches. In the meantime, the TTS-to-target conversion model can be also trained using parallel training corpus between the TTS and target speeches. The source-to-target conversion is realized by concatenating source-to-TTS and TTS-to-target models together as shown in Figure 4.5.

The proposed technique offers a novel scheme for the text independent voice conversion task. It does not require parallel training data as the model training is done on the two concatenated parallel trained models. The proposed solution can be extended and used in all applications of the text independent voice conversion where a TTS is readily available. The technique offers many benefits on the usability, for example it allows the training of voice conversion systems to use freely selected speech material (e.g. celebrity voices from TV, radio or movies).

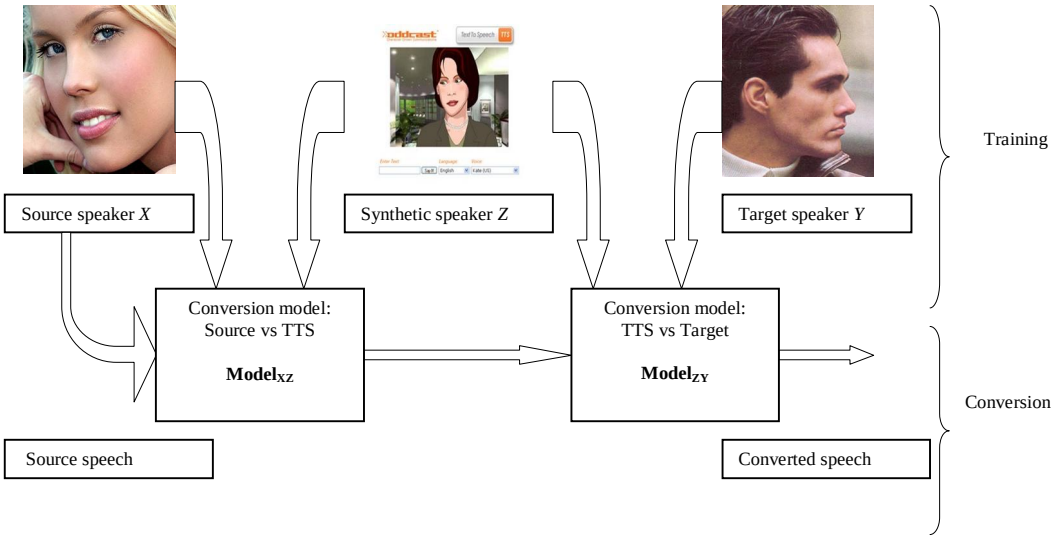

Figure 4.5: Diagram of the proposed text independent voice conversion system. (from [42])

The proposed method

The proposed technique can be realized in many ways. In this subsection, we only introduce one possible reference method designed for the text independent voice conversion application but it should be noted that this method is only exemplary. In all the alternatives, the core of the technique is that the text independent voice conversion models can be generally made by two concatenated voice conversion models trained with parallel utterances using a TTS system as intermediate step to bridge the non-parallel gap between the source and target speakers.

First, let us assume that we have a high-quality TTS system ready for use. Based on the training utterances from both the source and target speakers, the parallel synthetic speeches can be made available to the utterances of the source and target speakers.

To realize the main idea of the technique, we train first a model of the conversion between the source speaker X and the TTS voice Z using the parallel data of X and Z. Next, a second model is trained for converting the TTS voice Z to the target speaker's voice Y based on the Z-Y parallel data. In both cases the DTW is used to align the parallel training data and the aligned data is then used to train joint GMMs following the conversion approach introduced in [124] and presented in subsection 2.4.2. The joint GMMs are denoted $N_{XZ}(v, \mu, \Sigma)$ and $N_{ZY}(v, \mu, \Sigma)$ for the conversions X to Z and Z to Y, respectively. The symbol v is the joint input-output vector of the conversion, while μ and Σ represent the mean vectors and the covariance matrices of the GMM model.

Finally, the conversion model between non-parallel utterances from the source speaker X and the target speaker Y can be derived by concatenating two conversion functions defined from the GMM models $N_{XZ}(v,\mu,\Sigma)$ and $N_{ZY}(v,\mu,\Sigma)$ trained on parallel utterances. The conversion function that converts source feature x_t to target feature y_t is given by:

$$z_t = F_{XZ}(x_t) = \sum_{l=1}^{L} p_l(x_t) \cdot (\mu_l^z + \Sigma_l^{zx}(\Sigma_l^{xx})^{-1}(x_t - \mu_l^x)) \quad \text{where} \quad p_l(x_t) = \frac{c_l \cdot N(x_t, \mu_l^x, \Sigma_l^{xx})}{\sum_{i=1}^{L} c_i \cdot N(x_t, \mu_i^x, \Sigma_i^{xx})} \tag{4.17}$$

$$y_t = F_{ZY}(z_t) = \sum_{l=1}^{L'} p_l'(z_t) \cdot \left(\mu_l^y + \Sigma_l^{yz}(\Sigma_l^{zz})^{-1}(z_t - \mu_l^z)\right) \quad \text{where} \quad p_l'(z_t) = \frac{c_l' \cdot N(z_t, \mu_l^z, \Sigma_l^{zz})}{\sum_{i=1}^{L'} c_i' \cdot N(z_t, \mu_i^z, \Sigma_i^{zz})} \tag{4.18}$$

In the equations above, $p_l(x_t)$ and $p_l'(z_t)$ denote posterior probabilities of the l^{th} Gaussian component given the observations x_t and z_t and the GMMs $N_{XZ}(v,\mu,\Sigma)$ and $N_{ZY}(v,\mu,\Sigma)$, respectively; μ_l^z and μ_l^y are parts, corresponding to the output semi-space, of the joint center vectors of the models $N_{XZ}(v,\mu,\Sigma)$ and $N_{ZY}(v,\mu,\Sigma)$, respectively; Σ_l^{zx} and Σ_l^{xx} are blocks of the l^{th} covariance matrix of model $N_{XZ}(v,\mu,\Sigma)$ representing the output-input cross covariance and the auto-covariance of the input, respectively; Σ_l^{zy} and Σ_l^{zz} are defined similarly for the model $N_{ZY}(v,\mu,\Sigma)$; $F_{XZ}(.)$ and $F_{ZY}(.)$ denote the conversion functions between the source speaker X and the TTS speaker Z and between the TTS speaker Z and the target speaker Y, respectively; $N(x,\mu^x,\Sigma^{xx})$ and $N(z,\mu^z,\Sigma^{zz})$ represent sub-models of $N_{XZ}(v,\mu,\Sigma)$ and $N_{ZY}(v,\mu,\Sigma)$, respectively, corresponding to the input semi-space.

Discussion

The proposed technique has some important advantages. First, the method offers improved usability by eliminating the requirement for parallel data. The users can record their speech freely and that speech can be effectively used as training data. This aspect is crucial for practical applications. Secondly, the method is simple to implement being based on conventional and mature parallel voice conversion techniques. It is also possible to consider the proposed method as a solution for embedded applications.

On the other hand, several aspects of the proposed technique deserve some closer consideration. A high-quality TTS system is required during the training phase. This will add more memory and computational complexity. However, training can be carried out offline, and conversion does not need TTS system to support. Yet another aspect is that the system has to be tuned carefully since the cascaded/concatenated structure sometimes can accumulate the distortion.

Chapter 5

Contributions to GMM Framework

The conversion methods based on Gaussian mixture models (GMM) have been the most popular approach in the voice conversion literature. The data is modeled using a GMM and converted by a function that is a weighted sum of local regressions. In spite of its popularity, GMM based conversion is known to suffer from several drawbacks. One of them is common to all fitting tasks and is related to the model complexity requiring a trade-off between two objectives: accuracy of modeling and generalization capability. Simple models are subject to over-smoothing whereas complex models may result in over-fitting. A second drawback of GMM-based methods is related to the frame-based operation which ignores the temporal structure of speech. In addition to these fundamental problems further challenges may appear from particular use cases when, for example, the training data is limited. This chapter introduces a set of methods closely related to the GMM framework aiming to improve the modeling, to simplify the performance evaluation, to operate with limited data and to include the speech dynamics in the modeling.

The chapter is organized as follows. Section 5.1 presents an efficient scheme for the evaluation of GMM-based conversions based on the model parameters. In section 5.2, a conversion approach combining GMM modeling and a clustering scheme is presented. Section 5.3 describes a technique for adapting an existing well trained conversion model to a new target voice with a reduced amount of training speech. The chapter ends with concluding remarks presented in section 5.4. In addition, section 5.4 introduces a prospective GMM-based conversion technique which uses temporal dynamic features in addition to the static ones aiming to improve the frame-to-frame continuity. Parts of this chapter are based on [43][44][45][46].

5.1 MODEL EVALUATION SCHEME

Apparently, the quality of the trained GMM has a tremendous influence on the performance. Therefore, efficient objective evaluation of GMM models is becoming very important when going towards a better conversion quality. This section is based on [43][164] and presents a very efficient approach for the evaluation of GMM models directly from the model parameters without using any test data, facilitating the improvement of the transformation performance especially in the case of

embedded implementations. Though the proposed approach can be used in any application that utilizes GMM based transformation, we take voice conversion as an example application throughout the section. The proposed approach is experimented with in this context and compared against a MSE based evaluation method. The results show that the proposed method is in line with all subjective observations and MSE results.

The existing conventional objective approaches for GMM quality evaluation based on distance measures such as mean squared error require test or validation data and are rather heavy from the viewpoint of embedded implementations, which may prevent such quality evaluations in embedded applications. These kinds of approaches have several inherent drawbacks:

1. Memory and cost: need to obtain and store the validation data;
2. Consistency: test results are dependent on the selection of the test data, different results may be achieved using different validation sets;
3. Real-time feedback: difficult to integrate this kind of measurement into the model training process;
4. Complexity: computational load caused by the evaluation is rather high;

Thus more efficient objective GMM evaluation schemes that could avoid these problems should be investigated.

The approach presented in this section introduces a very efficient approach for objectively evaluating GMM quality that is readily suitable also for embedded implementations. The main idea in the proposed approach is to measure the quality of the model directly from the model parameters without using any test data. The approach makes it possible to generate better GMM models especially in practical embedded applications.

5.1.1 GMM Model Evaluation

The main idea presented in this section is to evaluate the quality of the GMM model directly based on the model parameters without using any testing data. More precisely, the measure utilizes the trace of target parts of the covariance matrices in the transform function to approximately evaluate the performance of GMM model in the transformation task. The proposed measure is very efficient to compute and it does not require any test data as the measurement is done directly from the model itself.

The evaluation method is derived by considering the properties of the GMM based transformation approach. The objective in the optimization of the GMM parameters in the conversion function is to minimize the average squared conversion error *D* for the training dataset.

$$D = \frac{1}{n} \cdot \sum_{t=1}^{n} \|y_t - F(x_t)\|^2 \tag{5.1}$$

where x_t and y_t denote aligned source and target vectors, respectively; n represents the length of the aligned source and target training data; t is a frame index and $F(.)$ represents the conversion function.

The mean squared error is usually computed also on a validation dataset to assess the GMM quality. Lower D scores indicate that trained GMM models perform better in the voice conversion task than a model with a larger D. Another approach for estimating the conversion error can be derived from data statistics (i.e., model parameters) using the variance of the distribution of y given x, i.e. $\varepsilon(x) = var(y|x)$. $\varepsilon(x)$ can be treated as a measure of the uncertainty of the conversion. The

smaller $\varepsilon(x)$ is, the more accurate the conversion performs. The proposed method originates from equation (5.1) and can be applied as an efficient measure for model assessment.

In theory, the quality of the GMM can be measured using:

$$Q = \int \varepsilon(x) \cdot p(x) \cdot dx \tag{5.2}$$

where Q represents a quality measure of the GMM and $p(x)$ is the probability density function of x.

To be able to estimate the quality from the model itself in practice, the different mixtures have to be taken into account in the computation. Moreover, to make the computational complexity lower, the following approximation is proposed, instead.

$$Q \approx \sum_{l=1}^{L} w_l \cdot tr\left(\Sigma_l^{yy}\right) \tag{5.3}$$

where $tr(.)$ denotes the trace of the matrix, w_l is the weight (prior probability) of the l^{th} component and Σ_l^{yy} is a block of the l^{th} covariance matrix describing the auto-covariance of the target semi-space. The value Q, also called trace measure and defined in equations (5.2) and (5.3), is proposed to be used for evaluation of GMM performance.

We have applied the GMM on the features in the discrete cosine transform (DCT) domain. The decorrelation tendency of DCT transformed features ensures almost diagonal covariance matrix. In this way the trace can better approximate the variance of the data in multiple dimensions. Therefore, equation (5.3) becomes more accurate. The GMM model performs better when Q value decreases. The proposed measure can be computed very efficiently and the measurement can be done directly on the model itself without any validation data. This measure can be used, for example for guiding the training of the transformation system towards better modeling. As very efficient implementations can be designed for the proposed scheme, it is particularly suitable for embedded applications. Nevertheless, the technique has the potential to benefit other applications as well, as it does not require any evaluation data and, given its design, is expected to produce consistent results.

5.1.2 Experiments

In order to verify the theoretical reasoning described in subsection 5.1.1, we carried out some experiments using voice conversion data. In these experiments, pitch and LSF parameters were studied mainly because of their importance in speech perception. Parallel utterances for two speakers were used for training (90 sentences) and testing (99 sentences). The LSF and pitch conversion models were trained on 20-dimensional joint LSF feature vectors and 2-dimensional joint pitch features, respectively, using the EM algorithm.

Trace measure vs. number of mixtures

A preliminary test was firstly carried out to verify that the proposed measure can meaningfully evaluate different models having different number of mixtures. Perceptual observations have indicated that the suitable number of mixtures for the conversion of LSFs and pitch features is 16 and 8, respectively, giving the best tradeoff between conversion quality and computational load. The trace measures for the corresponding models with different number of mixtures (as seen in Figure 5.1 and Figure 5.2) are completely in line with the perceptual observations.

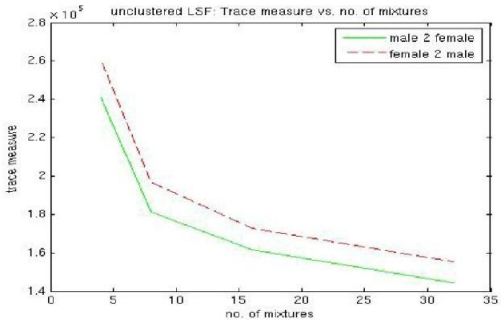

Figure 5.1: Trace measures vs. number of mixtures (LSF). (from [43])

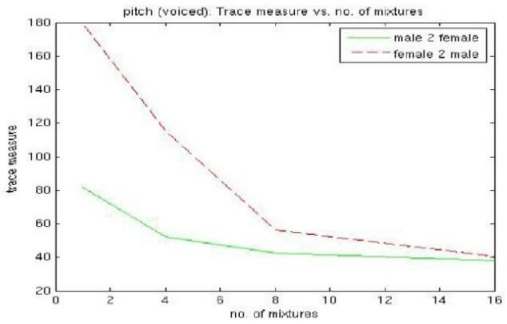

Figure 5.2: Trace measure vs. number of mixtures (pitch). (from [43])

Comparison between trace and MSE

The second experiment included comparative tests between the trace measure and the conventional MSE approach. Again, the evaluation included pitch and LSF parameters. The training was done on normalized data. Specifically, the features were first normalized using scaling. DCT transform is then applied to decorrelate the features. Accordingly, the conversion requires now normalization, DCT transform, mapping through GMM, inverse DCT transform and denormalization. It should be noted that the models were trained only on the training set, while both training and testing set were converted and analyzed separately for calculating MSE. Separate models were trained for the different directions of the conversion (from male to female and from female to male). GMM models are also trained on the voiced and unvoiced data, as denoted as model 1 and model 2. The converted data was compared to the target data in terms of MSE. The results from this experiment are given in Table 5.1:

Table 5.1: GMM models evaluated using MSE.

	GMM models	Female to Male	Male to Female
Test set	Pitch (voiced)	212	95
	LSF model 1	17438	16515
	LSF model 2	18213	16931
Train set	Pitch (voiced)	224	91
	LSF model 1	17199	16234
	LSF model 2	18050	17054

The trace measures of the same models are given in Table 5.2. They were computed using equation (5.3).

Table 5.2: GMM models evaluated using trace measure.

GMM models	Female to Male	Male to Female
Pitch (voiced)	0.785	0.473
LSF model 1	4.764	4.609
LSF model 2	5.029	4.886

As can be seen, the MSE and trace measures are completely in line with each other for both the training and validation sets. Moreover, the proposed measure can again also confirm our perceptual

findings on the voice conversion data: male-to-female conversion has better quality (smaller errors) than female-to-male conversion, and LSF model 1 outperforms LSF model 2.

5.1.3 Conclusions

In this section, we focused on the model evaluation aspects in the context of Gaussian mixture modeling based transformation. More specifically, we developed a novel procedure for efficient evaluation of the GMM models without using any evaluation data. The proposed approach was evaluated in the voice conversion task.

It is remarkable that the proposed trace measure is perfectly in line both with perceptual observations and MSE results (for both the training and validation sets). The use of the presented measure leads to the same conclusions with significantly less computation and without any validation data or perceptual evaluation. Thus, based on the presented practical experiments, the proposed trace measure can be regarded as an effective and efficient quality measure of the GMM model in transformation task.

The proposed GMM evaluation scheme offers several advantages when compared to the conventional MSF based evaluation technique:

1. Efficiency: very fast computation;
2. Simplicity: no validation/testing data needed for the evaluation;
3. Consistency: MSE results depend on the test data, but the trace measure always gives the same result provided the GMM is kept unchanged;
4. Easy integration: it is easy to integrate the analytical evaluation as a feedback into the model training, aiming to improve the models;

Consequently, it can be concluded that the proposed approach offers a very good solution for the evaluation of GMM model in the transformation applications. The method offers benefits in all implemented platforms, especially strong in embedded applications.

5.2 CLUSTERING AND MODE SELECTION

Since the statistical properties of speech signals depend heavily on the content, it is hard to design speech processing techniques that would perform robustly well on all inputs. In voice conversion, the type of inter-speaker relationship may differ depending on the type of speech segment. For example, nearly stationary voiced regions should usually be treated differently than plosives. For the problem of handling different types of speech segments, the solutions proposed in the literature include the use of acoustic similarity based classification and regression trees [53], phoneme-tied codebooks [118], K-means based clustering [165], and phoneme-based modeling [166].

To tackle this problem in a robust manner, we introduce a novel scheme for data clustering and mode selection. This technique has been previously described in [44][167]. The main idea of the proposed approach is to first cluster the target data to achieve minimized intra-cluster variability. A separate conversion scheme is trained and used for each cluster considering also the aligned vectors of the source speaker. According to the findings presented earlier in section 5.1 the reduction of target features variability is expected to lead to improved mapping accuracy if the clusters are identified correctly during the conversion phase. Moreover, the intra-cluster data behavior is less complex and thus, easier to model. Since the target data is missing in the conversion stage, the correct cluster or mode has to be recognized using only source-related information. Therefore, a

mode selector or classifier is trained on source-related data aligned to the clustered vectors of the target speaker in order to recognize the target-based clusters. Auxiliary speech features can be used to enhance the classification accuracy, in addition to the source data. The proposed scheme is fully data-driven and it avoids the need to use heuristic solutions. The superior performance of the proposed scheme has been verified in a real voice conversion system.

5.2.1 Proposed Approach for Data Clustering and Mode Selection

The voice conversion system used here has been briefly introduced in section 3.1.2. The conversion of the speech parameters is generally handled one frame at a time following the joint GMM based approach described in section 2.4.2. However a different GMM is trained from a separate cluster of training data and this section presents in detail the data clustering and mode selection scheme.

The proposed approach for data clustering and mode selection starts from the idea that the data should preferably be clustered into different operating modes in a way that is optimal for the desired goal. If the goal is to minimize the potential conversion error, the most effective approach would be to cluster the joint data from the source and the target side into different processing modes based on the target features. Due to the minimal intra-cluster variation of the target vectors, this choice ensures a minimized potential conversion error within each mode or cluster according to the findings presented in section 5.1. In this case, such a best possible clustering is based on target data that is not available during usage (conversion phase). During the conversion phase, the lack of target vectors introduces some uncertainty in the mode selection. In order to deal with this inconvenience, the next step is to train a mode selector that aims to find the correct cluster based on the data that is available during usage. This data can include in addition to the conventionally available data any auxiliary features that can be made available. Finally, a separate processing scheme is trained and used for each mode.

The proposed approach first finds M clusters solely based on the target data features y. For example, if the aim is to convert LSF vectors, the initial clustering is performed based on target LSF vectors only. The clustering can be performed e.g. using the well-known K-means algorithm to obtain the clusters $y^{(1)}, y^{(2)}, \dots, y^{(M)}$. When choosing the value of M care should be provided that each cluster receives sufficient data in order to avoid over-fitting and ensure a reliable training of the conversion model. Provided that each cluster has sufficient data, a larger number of clusters M should lead to a better accuracy. For the purpose of our discussion M can be empirically selected and does not need to be optimal.

After obtaining the initial grouping, the next step is to train a mode selector with the aim to recognize the target based clusters using only data from the source speaker. To facilitate the classification task, auxiliary features derived from the source data can be used in addition to the source vectors x. In principle, this auxiliary data denoted as aux can include any/all the features that one can *extract from the source data*. For example, the auxiliary feature set could include acoustic parameters such as pitch, voicing and energy as well as other parameters such as phoneme information, linguistic location, linguistic duration and part-of -speech. Given the initial target-based clusters, the extended aligned data set, denoted now as $z = [x^T\ aux^T\ y^T]^T$, can be straightforwardly split into the same M groups, $z^{(1)}, z^{(2)}, \dots, z^{(M)}$. Based on this grouping, it is possible to train a classifier aiming to find the correct cluster using only the source related data vector $[x^T\ aux^T]^T$. We implemented the classifier using a simple linear discriminative function $D([x^T\ aux^T]^T)$ but it would

also be possible to use other techniques such as non-linear discriminative functions, neural networks or support vector machines. The exact selection of the auxiliary feature set is not a highly critical issue in the sense that the features with no additional discriminative information will receive a very low or even a zero weight in the training while the more relevant features will receive a larger weight.

Once the mode selector or the classifier is available, a separate conversion scheme is trained for every mode with a training data set belonging to that mode. It is possible to either use the training data sets based on the initial clustering that was made based only on the target data or to re-cluster the data using the trained classifier to obtain re-grouped training data sets. The latter approach provides enhanced robustness against classification errors, and thus it should preferably be followed in cases where the classification error rate is not very low.

During the usage of the multi-mode processing system, the conversion system must first obtain the source vector to be converted and the corresponding auxiliary vector. This data is used as an input to the classifier that selects the mode. Finally, the conversion of the vector is handled using the conversion scheme corresponding to the selected mode.

The proposed approach summarized below has many beneficial properties. First, the approach is fully data-driven. Secondly, the method is very flexible in the sense that it is for example very easy to change the number of modes/clusters. Thirdly, there is no requirement to utilize any linguistic information but if such information is available it can easily be used to support the mode selection. Finally, the proposed method offers good performance as demonstrated in subsection 5.2.2 using practical experiments. The good performance can also be explained from another point of view. Figure 5.3 depicts the distribution of the first two frequencies of a LSF vector for a small set of target LSF vectors, selected randomly from a larger voice conversion training set. The line illustrates the ideal boundary for a two class clustering using K-means, whereas the circles and crosses represent the clustering decisions based on the widely used voiced/unvoiced classification. Provided that the mode selection is made correctly, it is evident that in the case of optimal clustering the distribution of any conversion errors will be much narrower than in the case of voiced/unvoiced clustering. While it is in general not possible to achieve a 100% mode classification rate, the proposed approach still successfully mimics this optimal case, leading to clear measurable improvements.

Training Algorithm:

Step 1: Define and extract an auxiliary feature set aux from the source training data set;

Step 2: Align the source related data and the target data to form extended combined feature vectors $z = [x^T\ aux^T\ y^T]^T$;

Step 3: Split the target data y into M clusters using e.g. the K-means algorithm;

Step 4: Group the extended vectors z into the same M clusters based on the clustered target data y;

Step 5: Train a mode classifier that aims at finding the correct target based cluster using only the source related features x and aux;

Step 6: Train M separate models for the different modes. Use as the training data the data classified to the corresponding cluster;

Conversion Algorithm:

Step 1: Extract the auxiliary feature vector aux from the source data;

Step 2: Select the correct mode using the source related vectors aux and x as input;

Step 3: Use the selected model to convert the source feature vector x;

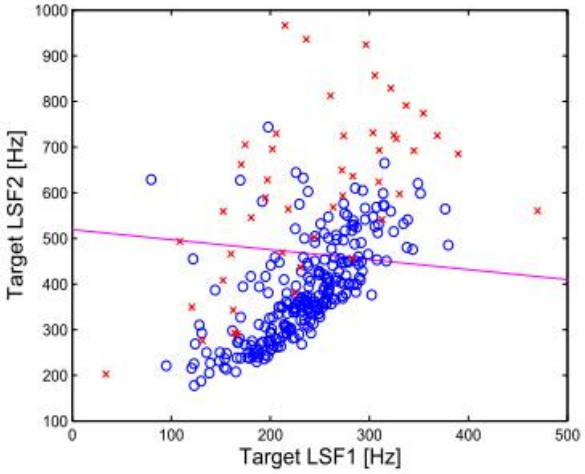

Figure 5.3: Ideal clustering vs. voiced/unvoiced clustering. The line illustrates the division between the two ideal clusters while o and x denote voiced and unvoiced data, respectively. It is easy to see that there is significantly less variability within each cluster in the case of ideal clustering. (from [44])

5.2.2 Experimental Results

The proposed data clustering and mode selection approach was tested in a voice conversion system that uses GMM to model the intra-cluster correspondence between the source and target vectors. To highlight the performance advantage achievable using the proposed method, we compared it against a common approach based on voiced/unvoiced clustering that also offers good performance. The comparison was done through the measurement of the average mean squared error between the converted and the target vectors.

Test set-up

The two different conversion schemes, the first based on the proposed approach and the second based on the traditional voiced/unvoiced clustering, were implemented for the conversion of LSF vectors. In the implementation of the proposed approach, we used several source-speech related features to form the auxiliary data vector aux. More specifically, the auxiliary data included the first and second derivatives of the LSF vectors, the pitch parameter, the energy parameter, the residual amplitude spectrum and the voicing information for the spectrum. The mode selector was implemented using a simple linear discriminative function. In the case of the traditional voiced/unvoiced clustering, we made a single voicing decision for each frame based on the voicing information for the spectrum.

Both conversion schemes were trained and tested using the same training and testing data sets. The speech data was selected from a TC-Star corpus of parallel English utterances [151]. A data set containing 90 sentences (29 880 frames) from a source speaker and a target speaker was used for the training while a distinct set of 99 sentences (32 700 frames) was reserved for the testing phase. In both sets, the source and target vectors were aligned using dynamic time warping supported with phoneme-level segmentation. All the conversions were handled using the joint Gaussian mixture modeling based approach summarized in subsection 2.4.2. One 16-mixture GMM was trained for each mode. In the proposed technique the number of modes is $M=2$. This value is not meant to be

optimal from the viewpoint of the tradeoff between accuracy and over-fitting but ensures a meaningful comparison to the other approach which uses two modes as well, voiced and unvoiced.

Since the mode classifier was implemented in a very simple way, we also implemented a third conversion scheme that directly utilized the perfect clustering based on the target data. Generally, the implementation of such a perfect classifier is not possible primarily because the training data is not available during the conversion phase. Nevertheless, this third conversion scheme can be used for demonstrating the theoretical performance bound that cannot be exceeded with the proposed approach provided that the initial clustering and the conversion schemes are kept unchanged.

Results

The results achieved in the test are summarized in Table 5.3. For the scheme based on the conventional voiced/unvoiced clustering, the mean squared error between the converted LSFs and the corresponding target LSFs was 23058 for the training set and 23559 for the testing set. For the proposed scheme, when implemented as described above, we achieved the MSE scores of 21015 and 21833 for the training set and the testing set, respectively. Since our classifier was implemented in a very simple way, we also tested the performance in the ideal hypothetical case with 100% classification rate. In this ideal case, providing the performance bound for the given initial clusters, the MSE figures were found to be 15295 and 15770 for the training and the testing set, respectively.

As is clearly evident from the results, the proposed method outperforms the conventional approach with a clear margin, despite the fact that the simple mode classifier only achieved a classification error rate of 12.4% (over the testing data). Moreover, the performance advantage was achieved even though the traditional voiced/unvoiced classification used as a reference also offered a very natural and efficient clustering scheme.

Table 5.3: Comparison between the conversion MSE achieved using the conventional voiced/unvoiced clustering and the proposed data-driven clustering schemes.

	Training Set	Testing Set
Voiced/unvoiced clustering	23058	23559
Proposed	21015	21833
Proposed (perfect classifier)	15295	15770

5.3 EFFICIENT RE-ESTIMATION

Gaussian mixture model (GMM) based techniques have been found to be efficient in the transformation of features represented as scalars or vectors. However, reasonably large amount of aligned training data is needed to achieve good results. To solve this problem, this section presents an efficient model re-estimation scheme [45]. The proposed technique is based on adjusting an existing well-trained conversion model for a new target speaker with only a very small amount of training data. The experimental results provided in the section demonstrate the efficiency of the re-estimation approach in line spectral frequency conversion and show that the proposed approach can reach good performance while using only a very limited amount of adaptation data.

GMM models in the voice conversion task are commonly trained from scratch using a relatively large amount of aligned training data. The training data can be either parallel, meaning both the source and target speakers read the same text, or non-parallel. Using a fairly large amount of data

improves the quality of the models, but this approach has several inherent drawbacks from different aspects:

- Usability: there is a need to record plenty of training data;
- Memory: requirement of having more memory available for storing the training data;
- Complexity: the computational load caused by the GMM training with large training data may be rather high.

Related adaptation techniques for GMM conversion with reduced data are described in [114] [129] and [168]. Maximum a posteriori (MAP) based adaptation [114] [129] trains a GMM on a large source corpus and limited amount of speech from the target speaker to estimate a joint GMM robustly. The eigenvoice conversion [168] uses multiple parallel sets of the same source speaker but several different target speakers to build a so called EV-GMM. Basically a GMM distribution is represented as a function of a weight vector. We can derive conversion functions to any target speaker using maximum likelihood techniques to estimate the weight vector from a reduced (and non-parallel) target data.

The re-estimation approach proposed in this section is applicable in a voice conversion framework where the source and target acoustic spaces are jointly modeled as a GMM. It introduces a very efficient scheme for adapting a well-trained GMM conversion model to a completely new target speaker with only limited speech data from the new target speaker. It is readily suitable for embedded implementations and it does not require a large amount of training data or data from many speakers. The proposed approach broadens the variety of potential use cases for voice conversion especially in practical embedded applications.

5.3.1 Efficient GMM Re-Estimation

In the GMM based transformation, multiple mixtures of Gaussian distributions are trained using joint feature vectors combined from the feature vectors estimated from the source and target sides. The main idea of the proposed re-estimation approach is to utilize an existing well-trained GMM model from the source speaker X to the target speaker Y (trained using an adequate amount of speech data), and to adapt it to be a GMM model from the source speaker X to a new target speaker Z with only a very limited amount of training data.

The proposed technique is very fast in adapting the voice conversion model to the new source and target speaker pair. It does not require much training data as the parameter estimation is done directly on the well-trained model. One possible use case is individualization of the text-to-speech functionality. With only a small amount of training data, TTS can start using any new voice provided by the user. The performance of the proposed approach is demonstrated using experimental results in section 5.3.2.

GMM training for source and target speakers: *X, Y*

First, let us assume that we have an adequate amount of training data from the source speaker X and the target speaker Y. In the case of the TTS application, it is rather easy to collect this speech data since a lot of speech is automatically available in the databases of the speech synthesis system. Also in other applications, the recordings of the two voices (X and Y) can be done quite easily e.g. by the developer of the application. After we have the data available, we can train a model that converts

speech from the speaker X to sound like the speaker Y just like in the case of conventional voice conversion.

The GMM for the random variable v can be estimated from a time sequence of v samples $[v_1\ v_2\ \dots\ v_t\ \dots\ v_w]$, when $v_k = [x_p{}^T\ y_q{}^T]^T$ is a joint variable and x_p, y_q would denote aligned features from the source X and target Y speaker respectively. The distribution of v is modeled by GMM as in equation (2.16) of section 2.4.2. The weighting terms are chosen to be the conditional probabilities that the feature vector v_t (at time t) belongs to the different components.

Re-estimation for source and target speakers: X, Z

As mentioned above, we have trained a GMM model, $\lambda\{c_l, N(v, \mu_l, \Sigma_l)\}$, for source speaker X and target speaker Y. For the new conversion pair from source speaker X to target speaker Z, given a limited training data of the joint variable $v_k = [x_p{}^T\ z_r{}^T]^T$, the GMM model can be adapted into $\hat{\lambda}\{\hat{c}_l, N(\hat{v}, \hat{\mu}_l, \hat{\Sigma}_l)\}$ for mapping between the source and the target speakers X, Z, based on the well-trained model λ.

Since c_l is the prior information to measure the probability that the training data falls into the cluster or the mixture, for the re-estimated GMM model $\hat{\lambda}$, the new target data does not change the data distribution for the source side. Thus it is reasonable to assume that the clusters have not changed for the source data. The outcome of having new target speaker only causes the cluster shifting along the target space as illustrated in Figure 5.4.

With the GMM model re-estimation scenario, if we assume that the model can be re-estimated only with the mean, while keeping the prior probability unchanged, we have the model re-estimation algorithm shown in equations (5.4) and (5.5).

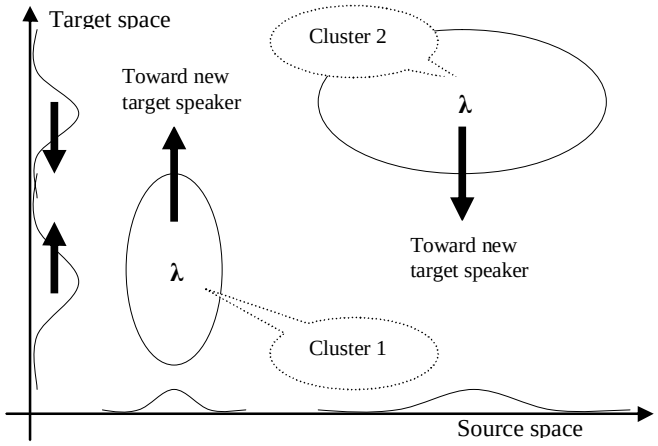

Figure 5.4: Diagram of GMM model re-estimation scenario. (from [45])

$$\begin{gathered} \hat{c}_l = c_l \\ \hat{\Sigma}_l = \Sigma_l \\ \hat{p}_l(x_t) = \frac{\hat{c}_l \cdot N(x_t, \mu_l^x, \Sigma_l^{xx})}{\sum_{i=1}^{L} \hat{c}_i \cdot N(x_t, \mu_i^x, \Sigma_i^{xx})} = p_l(x_t) \end{gathered} \tag{5.4}$$

$$\begin{gathered} \hat{\mu}_l^z = \frac{\sum_{m=1}^{M} \hat{p}_l(x_m) \cdot z_m}{\sum_{m=1}^{M} \hat{p}_l(x_m)} \\ \hat{\mu}_l = [(\mu_l^x)^T\ (\hat{\mu}_l^z)^T]^T \end{gathered} \tag{5.5}$$

where $c_l, \hat{c}_l$ represent prior probabilities of the l^{th} mixture component in the original and adapted joint GMM; $\Sigma_l, \hat{\Sigma}_l$ denote the covariance matrices, $\mu_l, \hat{\mu}_l$ the mean vectors and $p_l(x_t), \hat{p}_l(x_t)$ the posterior probabilities of the l^{th} component for the original and adapted joint GMM, respectively; $\mu_l^x, \hat{\mu}_l^z$ represent halves of the joint vectors μ_l and $\hat{\mu}_l$ corresponding to the speakers X and Z, respectively. In equation (5.5) z_m represents a feature vector of the new target speaker aligned with source speaker's vector x_m.

Depending on the size of the training data available from the target side (new target Z), we can also estimate the covariance matrix as shown in equation (5.6).

$$\hat{\Sigma}_l = \frac{\sum_{m=1}^{M} \hat{p}_l(x_m) \cdot ([x_m, z_m] - [\mu_l^x, \hat{\mu}_l^z]) \cdot ([x_m, z_m] - [\mu_l^x, \hat{\mu}_l^z])^T}{\sum_{m=1}^{M} \hat{p}_l(x_m)} \tag{5.6}$$

If the size of the training data is very small (less than roughly 5 utterances or 15-20 seconds of speech), it is reasonable to keep the covariance matrix unchanged due to the following reasons:

- The covariance matrix cannot be reliably estimated with very limited amounts of data;
- The source space has not changed at all;

The contribution of the covariance in voice conversion is very small due to the small weighting factor in $\Sigma_l^{yx}(\Sigma_l^{xx})^{-1}$.

5.3.2 Experimental Results

We carried out several experiments to demonstrate that a well-trained GMM converting voice X to voice Y can be effectively adapted to a new target Z using a very limited amount of parallel X to Z adaptation data. Both objective and subjective measures were used in the experiments.

10^{th}-order LSF vectors were used in our experiment. Based on our previous experiments [36], the transformation of LSFs can be handled using a GMM of 8 mixture components and thus the models discussed below use 8-component GMMs. It was observed that reducing the number of components degrades the modeling. In turn, adding components only brings marginal improvement while the complexity increases and the parameter estimation becomes less reliable. Moreover, we had a parallel corpus of speakers X, Y and Z available where X and Y are female voices and Z is a male. Speaker Z was selected to have a different gender than X and Y to make the situation more challenging for the new scheme. Altogether 126 utterances from each speaker grouped formed the training set while there was also a separate test set of 10 sentences available from each speaker. The speech data consisted of UK-English samples with a sampling rate of 8kHz taken from a TC-Star corpus [151].

In the first experiment summarized in Table 5.4, the performance of voice conversion was evaluated between the conventional approach using full training set and re-estimation approach using very limited adaptation data, more precisely 1 or 3 utterances. No specific criterion was used for the selection of those (1 and 3) utterances but phonetically balanced utterances are preferred for optimal results. For the conventional approach, a baseline conversion model, denoted as Baseline, was trained on the full training set from source X to target Z to demonstrate the upper performance limit. For the re-estimation approach, we firstly trained a seed conversion model GMM^{XY} on the full training set from source X to target Y. A parallel subset of 1 or 3 utterances from speakers X and Z was used to adapt the seed conversion model using re-estimation algorithm mentioned above, resulting in the corresponding GMM model, denoted as Adapt. The model re-estimation was performed using only the mean parameters.

In addition, small subsets of training data (1 or 3 utterances) were also used to build models between X and Z from scratch by EM training (denoted as EM). These models can be compared with the re-estimated conversion model using the same amount of data as seen in Table 5.4. The table shows the mean squared error (MSE) between converted and target speech (LSF vectors represented in Hz), measuring the performance in an objective way.

Modified mean opinion score (MOS) tests were also carried out to provide a numerical indication of the perceived conversion performance of the conventional and the re-estimation based approaches in a subjective listening test. The score was expressed as a single number in the range -2 to +2, where 0 denotes identical conversion performance, and +/-2 indicates a large difference in the perceived conversion performance. When comparing "A vs. B", +2 favors A and -2 favors B. The samples were evaluated on two criteria, the identity match and the converted speech quality but only one number was assigned as a subjective score of preference with these two criteria in mind.

Table 5.4: MSE between the converted and the target (Z)

	Adapt	EM	Baseline
1 utt	21630	27491	16284
3 utts	20460	21052	16284

Table 5.5: Subjective listening test between baseline and adaptation approaches using 1 utterance.

Adapt vs. EM	Adapt vs. Baseline
+1.65	-1.0

In the subjective listening tests summarized in Table 5.5 a set of 10 utterances was evaluated by 11 testers and the average score of this evaluation was 1.65 in the favour of Adapt against EM, and -1.0 favouring Baseline against Adapt as shown in Table 5.5. Only the case of 1 training utterance is evaluated in the listening test because it demonstrates the efficiency of the proposed scheme in the case when less data is available for the re-estimation. The result clearly indicates that the performance using re-estimation is not as good as Baseline trained with a large data set but it is reasonably close to it considering the fact that only one training sentence was used in the re-estimation. On the other hand, the proposed approach clearly outperforms conventional EM training with small data sets.

5.3.3 Discussion

It should be noted that voice conversion consists of both excitation and LSF conversion, and thus, strictly speaking, the objective measurements done on LSFs and the listening test results do not measure the same aspects. Nevertheless, the experiments supported each other.

If the data set is extremely small, the covariance estimation becomes highly unreliable. It has been observed that the performance degrades if the re-estimation is done on combined mean and covariance as compared to mean-alone re-estimation. The covariance information becomes beneficial in re-estimation if the data is sufficient. The contribution of the covariance re-estimation may be important if the estimation is reliable.

A major strength of the proposed scheme is that it takes advantage of the re-estimation of the baseline model with only a limited amount of adaptation data. With the conventional EM based solution, it is much more difficult to have reliable prior and covariance information since they cannot

be reliably estimated from the reduced data. As a consequence, there are many practical advantages of the proposed approach as listed below.

- Only a very limited amount of speech data needs to be recorded. This is crucial for many practical embedded applications;
- Low complexity: less computation to adapt the model, no need to train on full training sets;
- Efficiency: fast model adaptation;
- Ideal solution especially for embedded applications;

5.4 CONCLUSIONS

This chapter tackled different challenges of GMM-based voice conversion aiming to improve the conversion accuracy but also to increase the versatility of the framework or ease the evaluation process. First, an evaluation procedure for the GMM-based transformations was presented. The procedure does not require evaluation data but computes an accuracy measure from the GMM parameters instead. The experiments indicate that the proposed measure is fast and in line both with perceptual observations and with MSE objective results.

In order to improve the mapping accuracy, a novel scheme for clustering and mode detection was proposed in section 5.2. Starting from the finding that the accuracy of the GMM-mapping is directly related to the data variance, the proposed idea is to cluster the data such as to minimize the intra-cluster variation. Then a mode selector is trained in order to find the correct cluster based on the data available during the conversion phase. An auxiliary vector derived from the source data was used to improve the discrimination capability of the mode classifier. Finally, a separate GMM is trained for each cluster. The experimental results show that the proposed approach outperforms an equivalent conversion scheme that uses voicing based classification. In addition to the performance advantage, the proposed method is flexible and it enjoys the benefit of being a completely data-driven technique, eliminating the need for linguistic knowledge.

An existing well-trained conversion model between a pair of speakers can be adapted to a new target speaker even with a small amount of training data using the procedure described in section 5.3. As can be seen from both objective and subjective results presented in section 5.3.2, the experiments indicate that the adapted model performs clearly better than an equivalent EM model trained from the reduced data. In the evaluations it has been observed also that the re-estimation of covariance matrices may become unreliable for too small data. The mean re-estimation can be used effectively with extremely reduced data. The performance is not very sensitive to the amount of data and it is reasonably close to the baseline system. The experimental results demonstrate the efficiency of the re-estimation approach in LSF conversion.

The next subsection discusses problems related to continuity and temporal evolution of speech and proposes a future research direction. The proposed method adds dynamic features to the conventional static vectors in order to account for the temporal information and applies a regular GMM-based conversion. Both static and dynamic parts of the converted vectors are used to form an objective function whose optimization is expected to lead to improved naturalness of the converted speech.

5.4.1 Future Work Proposal: Enhanced Voice Conversion Using Temporal Dynamic Features

It is common in voice conversion that the trained model transforms the feature vector of a source speaker to a feature vector of a target speaker using frame-by-frame processing without modeling the temporal characteristics across frames [26][58][124]. In conventional voice conversion systems, the conversion models are trained using a set of paired static vectors parameterizing aligned frames of the source and target speech [18]. In the conversion phase, the feature vectors from the source speaker are transformed into the feature vectors of the target speaker using the trained models, and the converted speech is generated from a sequence of transformed static feature vectors. In spite of the fairly good performance of this methodology, in the absence of a model for the timing structure, there is a clear gap between conventional modeling and the natural speech, as timing is one of the most important features in the speech signal.

The problem of how to effectively handle temporal information or the structure across the frames has been addressed to some extent in the recent literature [100][121][133], and some of the proposed solutions have been discussed in section 2.4.2 in the context of GMM-based conversion. In Hidden Markov Model based voice conversion [53], temporal information is implicitly modeled in state transitions. In some prior art approaches, e.g. [44], dynamic features from the source side are used for guiding the modeling but the full exploitation of also target-side dynamic features during synthesis has not been discussed in the literature.

The method introduced in this section and originally presented in [46] aims to improve the temporal structure of the converted speech by taking advantage of dynamic features of both the source and the target speaker. The main idea of the technique is to extract dynamic features that are appended to the feature vectors and to use the dynamic information to improve the output at the synthesis time. The dynamic feature vector can be, for example, the first derivative of the original feature vector. The first derivative indicates the change tendency of the current parameter vector and is one of the simplest ways to describe the dynamics of the parameter tracks. The conversion models are then trained using frame-wise combined source-target vectors, including the dynamic features from both the source and the target. In the conversion phase, the generalized feature vector from source speaker is transformed to the generalized feature vector of the target speaker, including the dynamic features. The converted feature vector is re-estimated or rediscovered from a sequence of transformed static and dynamic feature vectors, and then used in the synthesis. The proposed technique has the potential to significantly improve the temporal structure and the quality of converted speech. In this section, the discussion is limited to the technical details of the technique leaving the full implementation and evaluation for future study.

The proposed method

Let $x = [x_1\ x_2 \dots x_t \dots x_n]$ be the sequence of static feature vectors characterizing a speech produced by the source speaker and $y = [y_1\ y_2 \dots y_t \dots y_n]$ be the corresponding aligned static feature vectors describing the same content as produced by the target speaker, where x_t, y_t are speech vectors at time t. Now, the dynamic feature vectors x_t' and y_t' at time t are appended to the static feature vectors to form generalized feature vectors,

$$x_t \rightarrow \begin{bmatrix} x_t \\ x_t' \end{bmatrix}, \qquad y_t \rightarrow \begin{bmatrix} y_t \\ y_t' \end{bmatrix}. \tag{5.7}$$

The dynamic feature vectors can be estimated using several different techniques that have different accuracy and complexity tradeoffs. For example, the dynamic features can be computed using an FIR filter. It is also possible to use a very approximate technique for estimating the first derivative of the original feature vector, in the simplest case as follows:

$$x'_t = \frac{dx_t}{dt} \approx \sum_{i=-p}^{q} a_i \cdot x_{t-i} \approx x_t - x_{t-1}\,, \qquad y'_t = \frac{dy}{dt} \approx \sum_{i=-p}^{q} a_i \cdot y_{t-i} \approx y_t - y_{t-1} \tag{5.8}$$

The conversion models are trained similarly as in the conventional approach, except that the feature vector is now generalized to include the dynamic feature vector. As a consequence, the converted feature vector is composed of static and dynamic parts of the converted feature vector;

$$\begin{bmatrix} c_t \\ c'_t \end{bmatrix} = F\left(\begin{bmatrix} x_t \\ x'_t \end{bmatrix}\right) \tag{5.9}$$

The final converted static feature vector $\hat{c}_t$ is re-estimated from c_t and c'_t by optimizing the objective function

$$\begin{aligned} Q &= (1-\lambda) \cdot \|\hat{c} - c\| + \lambda \cdot \|\hat{c}' - c'\| \\ &= (1-\lambda) \cdot \frac{1}{n} \cdot \sum_{t=1}^{n} (\hat{c}_t - c_t)^2 + \lambda \cdot \frac{1}{n} \cdot \sum_{t=1}^{n} (\hat{c}'_t - c'_t)^2 \end{aligned} \tag{5.10}$$

where $0 \leq \lambda \leq 1$ is a factor for balancing the importance of the static and dynamic features. The value of λ can be decided empirically by perceptual evaluation. The notation $\hat{c}'_t$ denotes the time derivative of $\hat{c}_t$. By minimizing the objective function Q, we can have the re-estimated converted static feature vector $\hat{c}_t$ either using an analytical solution by solving the equation group (5.11) or using an iterative numerical solution.

$$\frac{\partial Q}{\partial \hat{c}_t} = 0, \qquad t = 1, \dots, n \tag{5.11}$$

Finally the converted speech is synthesized from the re-estimated target static feature vectors $\hat{c}_t$. The synthesis can be performed using existing techniques.

In practice, an efficient algorithm has to be implemented to reduce the computational complexity of the optimization step. Intuitively, optimizing the criterion given in equation (5.10) can be regarded as balancing between the parameter tracks given by the converted static vectors and those dictated by the converted dynamic features. A possible approach towards an approximate solution to this optimization problem with very low computational complexity is proposed below.

The static features can be recovered by integrating (summing) over the dynamic features assuming that the initial value is known, otherwise the initial value should be estimated. We refer to this operation as dynamic-static transform and denote it as DS. DS transform can be implemented as:

$$\hat{c}_{r,t} = DS(\hat{c}'_t) = \alpha + \sum_{t} \hat{c}'_t \approx \hat{c}_{r,t-1} + \hat{c}'_t \tag{5.12}$$

where $\hat{c}_{r,t}$ represents the recovered static feature vector. The constant α is the integral bias corresponding to the initial value of the static vector $\hat{c}_{r,t}$. It can be estimated, for example, by minimizing equation (5.13).

$$\alpha_{opt} = \underset{\alpha}{\operatorname{argmin}} \sum_{t} \left\| c_t - \hat{c}_{r,t} \right\| \tag{5.13}$$

The re-estimated static feature can be efficiently calculated using

$$\hat{c}_t = (1-\beta) \cdot c_t + \beta \cdot \hat{c}_{r,t} \tag{5.14}$$

The factor β can be empirically obtained to balance between static and dynamic features. It can also be made adaptively, so that it can be adjusted depending on the quality of static and dynamic features along the time.

Advantages and disadvantages

The technique has a number of advantages. By taking into consideration the relationship between consecutive frames the proposed approach has the potential to improve the temporal structure and naturalness of the converted speech and to enhance the robustness. The method also offers a flexible mechanism to balance between the contribution of the static and dynamic features. Another advantage is that the technique is not limited to a particular voice conversion system and can be used with different systems.

Basically there is no significant drawback for the proposed idea. The memory overhead is very limited. With an efficient algorithm, there is only a marginal change in the computational complexity compared to the conventional approach for conversion, especially in the actual conversion phase.

Chapter 6

Alternative Model Estimation Techniques

Spectral conversion represents a central task of voice conversion and has been the main focus of the research in this area. The state of the art of spectral conversion techniques reviewed in Chapter 2 reveals that in spite of fairly successful results, additional advances are necessary towards excellent identity mapping and speech quality. It has been observed that the techniques are characterized in general by a tradeoff between the speech quality and the identity conversion. This can be partly explained by the increased degree of manipulation necessary to achieve a better identity mapping which inherently accumulates more degradation. On the other hand, there is a masking effect between these aspects since a better quality allows more sensitive identity differentiation. In this chapter the aim is to improve the spectral conversion first by researching into new conversion models and secondly by proposing enhancements and solutions to shortcomings of some exiting approaches. It is argued for instance that increased naturalness and better sound quality can be achieved by including the speech dynamics in the conversion model. It is also discussed that vector quantization offers a good foundation for conversion techniques more advanced than the conventional codebook mapping. The ideas presented in this chapter are partly based on studies published in [40] [47] [48] [49].

In section 6.1 the multi-stage vector quantization (MSVQ) technique from speech coding is proposed to obtain memory efficient mapping codebooks. In section 6.2 a new conversion method based on locally trained linear transformations is presented. Section 6.3 introduces a new conversion approach based on factor analysis and bilinear models. Finally, in section 6.4 the main conclusions are summarized and two directions for future research are proposed.

6.1 MEMORY EFFICIENT MSVQ FOR VOICE CONVERSION

The codebook based techniques [26][51][120][121] are a common approach that can be utilized in voice conversion. Although promising results have been reported, two important drawbacks remain: 1) the output often contains discontinuities, and 2) the memory requirements and the computational

complexity might become large if the target is to achieve accurate conversion results. The technique proposed in this section and based on [49] introduces a novel approach for codebook based voice conversion that alleviates these problems. Moreover, according to tentative tests, the method can also improve the conversion accuracy.

The basic idea of codebook based voice conversion was presented in [26]. A prior-art approach for improving the continuity was introduced in [51] [169]. The size of the trained codebook is typically a fraction of full size of the training data but the full data could also be treated as a codebook by itself in some implementations. The technique presented in this section brings significant improvements compared to these prior art solutions.

This section offers a novel method for codebook based voice conversion that both significantly reduces the memory footprint and improves the continuity of the output. Moreover, the method may also reduce the computational complexity and enhance the accuracy of the conversion. The footprint reduction is achieved by implementing the paired source-target codebook as a multi-stage vector quantizer (MSVQ). During conversion, the N best candidates in a tree search are taken as the output from the quantizer. The N candidates for each vector to be converted are used in a dynamic programming based approach that aims at finding a smooth but accurate output sequence. The method is flexible and it can be used in different voice conversion systems.

6.1.1 The Proposed Method

The degree to which spectral details are preserved by a codebook depends greatly on the codebook size. In order to cover the acoustic space with an average spectral distortion of 1dB a very large codebook is necessary leading to intractable complexity and huge memory requirements. Such storage and complexity problems can be addressed by organizing the codebook in a multi-stage structure as shown in Figure 6.1. We present the idea of multi-stage vector quantization [170] showing how it can be utilized in voice conversion and discussing its advantages in a voice conversion framework.

In a multi-stage VQ a feature vector z of dimension p is approximated as:

$$\begin{aligned} \hat{z} &= c_0^{(l_0)} + c_1^{(l_1)} + \cdots + c_{K-1}^{(l_{K-1})} \\ &= B_0^{(l_0)} c_0 + B_1^{(l_1)} c_1 + \cdots + B_{K-1}^{(l_{K-1})} c_{K-1} = Bc \end{aligned} \tag{6.1}$$

where $\hat{z}$ is the quantization of z, K is the number of stages and $c_k^{(l)}$ is the l-th codeword from the k^{th} stage. The k^{th} stage has a total of L_k code-words stacked as c_k (dimension $L_k p$ x 1).

$$\begin{aligned} c_k &= \left[c_k^{(0)^T}, c_k^{(1)^T}, \ldots, c_k^{(L_k-1)^T} \right]^T, \\ c &= [c_0{}^T, c_1{}^T, \ldots, c_{K-1}{}^T]^T \quad \text{and} \\ B &= \left[B_0^{(l_0)}, B_1^{(l_1)}, \ldots, B_{K-1}^{(l_{K-1})} \right]. \end{aligned} \tag{6.2}$$

$B_k^{(l)}$ is a sparse Toeplitz matrix with 0 and 1 elements used to select the l^{th} codeword from the k^{th} stage $c_k^{(l)} = B_k^{(l)} c_k$. For a multi-stage codebook the number of stages and their sizes can be adjusted depending on the design objectives formulated in terms of accuracy, memory footprint, computational complexity etc.

In voice conversion such a multi-stage vector quantizer can be trained from the joint vector sequence $[z_1, \ldots, z_t, \ldots, z_T]$ of aligned source and target vectors $z_t = [x_t{}^T, y_t{}^T]^T$ using a distortion

measure suitable for the joint acoustic space. The training can be done with the simultaneous joint design algorithm proposed in [170] which optimizes all the stages simultaneously.

In order to convert a source input sequence $[x_1, \dots, x_t, \dots, x_T]$ the search is performed only on the source side of the codebook but the target part of the search result is used as output.

The search procedure can be implemented as sequential search, typical approach, or as M-L search as proposed in [170]. Given the input vector x_t a typical sequential search would first find the closest codeword in the first stage $cx_0^{(l_0)}$ according to:

$$d\left(x_t, cx_0^{(l_0)}\right) \leq d\left(x_t, cx_0^{(l)}\right), \quad \forall l \neq l_0 \tag{6.3}$$

where $d(.)$ denotes a generic acoustic distance. Then, from the second stage, the codeword $cx_1^{(l_1)}$ closest to $x_t - cx_0^{(l_0)}$ is found based on:

$$d\left(x_t - cx_0^{(l_0)}, cx_1^{(l_1)}\right) \leq d\left(x_t - cx_0^{(l_0)}, cx_1^{(l)}\right), \quad \forall l \neq l_1 \tag{6.4}$$

and so forth. The sequential has the advantage of computational efficiency but its performance sensibly degrades when more than two stages are used because at each stage the existence of subsequent stages is ignored.

The M-L search depicted in Figure 6.1 is a tradeoff between the complexity of a full search (otherwise more accurate) and the performance degradation of a sequential search (more computationally efficient).

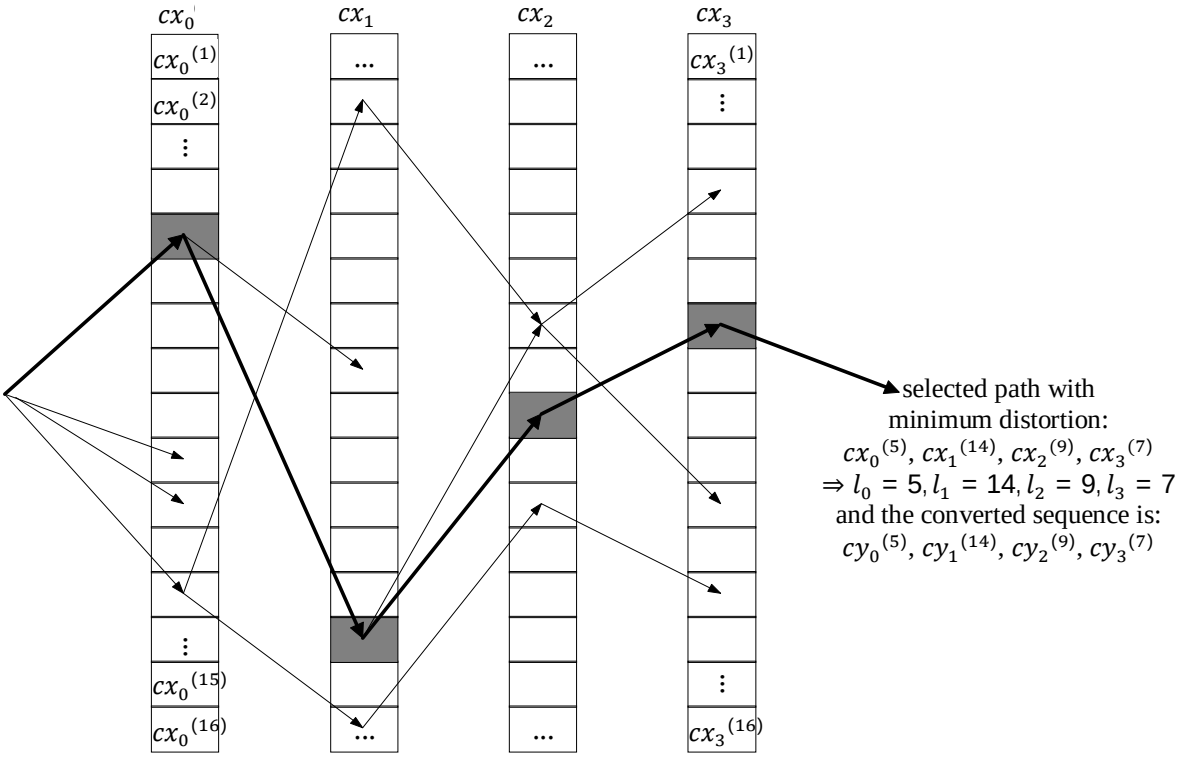

Figure 6.1: Example of M-L tree search in a 4-stage MSVQ. At each stage M best code-words are selected for each of M candidate paths from the previous stage and only M best paths are preserved. (from [49])

Starting from the input vector x_t, the M closest code-words $cx_0^{(l_0)}$ are selected from the first stage having the smallest distances $d(x_t, cx_0^{(l_0)})$ and the M difference vectors are calculated. For each difference vector, the M nearest code-words are selected from the second stage producing a total of

M^2 paths. Only M paths which achieve the lowest distortion are preserved while the rest are discarded and the process continues until the last stage K. This procedure ends up with M paths outputting M best candidates for the quantized representation of x_t. [171] indicates that M-L search can achieve accuracy close to that of a full search for relatively small values of M. Typically only the candidate (or path) with the smallest approximation error is interesting but the number N of best candidates considered can be set according to the design goals.

After the N best candidates are available for a given number of vectors to be converted, the optimized output sequence is obtained using dynamic programming. For each candidate, the corresponding source-space distance is stored during the search procedure. In addition, a transition distance is computed between each neighboring candidate pair accounting for the smoothness of the converted result. These distances together are used in the dynamic programming based approach for finding the "optimal output sequence", i.e. the path that gives the smallest overall distance. The relative importance between the accuracy and the smoothness can be set using weights.

6.1.2 Experimental Results

The method was tested in a practical voice conversion environment in the conversion of the line spectral frequencies (LSFs). The 10-dimensional LSF parameters were estimated from 90 sentences at 10 ms intervals. 14,942 vectors were selected for training and a distinct set of another 14,942 vectors were used for testing. The set of 90 sentences was selected from a TC-Star parallel corpus [151] of UK-English speech and sampled at 8 kHz.

The test included three set-ups. The first set-up (A) followed the proposed technique using three stages with 16 vectors in each stage. The second set-up (B) included a full codebook containing all the training vectors, while the third (C) contained a small codebook having the same footprint as the proposed set-up A (with real source-target vectors). In set-up (C) the codebook size is equal to the number of source-target vectors stored by (A) and the code-words are obtained from a K-means clustering by replacing the quantized vectors with the closest source-target vector pairs from the training data. The dynamic programming step was omitted to obtain comparable results.

The three methods were evaluated from three different viewpoints: performance/accuracy, memory requirements, and computational load. The accuracy was measured using the average mean squared error, while the memory requirements were computed as the number of vector elements that have to be stored in the memory, and the computational load is estimated as the number of vector comparisons required during the search procedure. The results of the evaluation, computed using the testing data, are summarized in Table 6.1. These results show that the proposed technique performs well with respect to all aspects: it produced clearly the best accuracy and the lowest memory figures. Method C offers similar memory and complexity levels but the conversion accuracy is much lower compared to A, therefore the proposed technique can be considered a clear winner in the evaluation.

Table 6.1: Comparison of three different techniques in terms of performance (accuracy), memory requirements and computational complexity.

	Proposed (A)	Baseline prior art (B)	Low-memory prior art (C)
Accuracy (MSE, $\cdot 10^4$)	3.62	4.12	4.79
Memory (number of vector elements)	960	298,840	960
Complexity (number of vector comparisons)	144	14,942	48

6.1.3 Advantages and Disadvantages

The benefits of multi-stage vector quantization in voice conversion include significant memory reduction, increased computational efficiency, flexibility and scalability to different design requirements, but also improved continuity and accuracy of the converted results [49]. The method is fully data driven and can be used to prevent over-fitting. There are no known disadvantages. Theoretically the MSVQ scheme can be easily combined with the ideas presented in the next section.

6.2 LOCAL LINEAR TRANSFORMATION

Many popular approaches to spectral conversion involve linear transformations determined for particular acoustic classes and compute the converted result as linear combination between different local transformations in an attempt to ensure a continuous conversion. These methods often produce over-smoothed spectra and parameter tracks. The method proposed in this section and published in [48] computes an individual linear transformation for every feature vector based on a small neighborhood in the acoustic space thus preserving local details. The method effectively reduces over-smoothing by eliminating undesired contributions from acoustically remote regions. The method is evaluated in listening tests against the well-known Gaussian Mixture Model based conversion, representative of the class of methods involving linear transformations. Perceptual results indicate a clear preference for the proposed scheme.

Many existing approaches involve linear conversion functions and typically suffer from two important drawbacks. One of them is related to the frame based operation in which the temporal continuity of the spectral features is ignored. The second issue is the so-called over-smoothing characterized by an undesired smoothing of the parameter tracks and converted spectra. The combined effect of these drawbacks is a poor speech quality.

The GMM based approach is very popular and representative of the class of methods based on linear transformations. In GMM based conversion, a linear transformation is trained for each Gaussian component and the result is computed as a weighted sum of local regression functions in an attempt to avoid sudden changes of the conversion function. In reality, a frame's decomposition is dominated by only one mixture component [99] making the method susceptible to discontinuities. In addition, the GMM technique is also affected by over-smoothing.

In this section we propose a spectral conversion scheme which trains an individual linear transformation for each feature vector. The method uses an underlying codebook trained from the aligned data of the two speakers and the linear transformation is computed on a selected set of codebook centers situated in the proximity of the input spectral vector in the acoustic space. By focusing on the local properties of the acoustic space, the proposed method is shown to effectively reduce the over-smoothing. Our listening tests suggest that the proposed scheme is probably affected to a lesser degree by discontinuity artifacts than the GMM approach.

While suffering serious limitations as a conversion method in itself, the codebook has the favorable property of good detail preservation which benefits the proposed algorithm where such limitations are avoided.

6.2.1 The Proposed Method

The use of linear transformation for spectral conversion is not new. An important number of solutions based on linear transformation have been proposed in the literature.

In [172] the aligned spectral vectors of source and target speakers are first divided into a number of classes and a linear transformation is trained for each class. All the linear transformations contribute to the conversion of each source vector in the form of a weighted sum where the weights represent probabilities that the source vector belongs to the corresponding class. The GMM based solution [124] works in a similar way using one linear transformation for each mixture component.

By analogy with [173], which argues that linear combinations over large sets of curves are bound to produce averaged results and destroy characteristic details, we believe that allowing all the linear transformations to contribute to the conversion is likely to produce a similar averaging effect equivalent to over-smoothing. Similar to Freeman et al. [173], we believe it would be beneficial to restrict the number of linear transformations involved in conversion to only a few corresponding to the most similar speech classes. In this section we take this idea forward and propose a local regression approach where each source vector is converted with an individual linear transformation trained locally within the neighborhood of the input vector. This method can be seen, in some sense, as a tradeoff between the mapping codebooks and, for instance, the traditional GMM approach.

Assume that our training set consists of two time aligned sequences of source and target spectral vectors, denoted X and Y, and let us consider the codebook M with Q centers obtained from the quantization of sequence Z of the combined vectors $z_n = [{x_n}^T \; {y_n}^T]^T$.

$$X = [x_1, x_2, \dots, x_N] \quad Y = [y_1, y_2, \dots, y_N] \tag{6.5}$$

$$M = \left[\begin{bmatrix}\mu_1^x \\ \mu_1^y\end{bmatrix} \quad \begin{bmatrix}\mu_2^x \\ \mu_2^y\end{bmatrix} \quad \dots \quad \begin{bmatrix}\mu_Q^x \\ \mu_Q^y\end{bmatrix}\right]. \tag{6.6}$$

The idea of local regression is to fit local models to nearby data. The conversion of a source vector x requires, in a first phase, the selection of a so called neighborhood of x or set of codebook centers situated in the proximity of x. The simplest way to determine the neighborhood of x is to consider its K nearest neighbors that minimize the Euclidean distance $d_E(.,.)$:

$$d_E(x, \mu_q^x) = \|x - \mu_q^x\| \tag{6.7}$$

The neighborhood can be expressed formally as:

$$N(x) = \{\mu_{q_1}, \mu_{q_2}, \dots, \mu_{q_K}\} \tag{6.8}$$

where q_k are codebook indices of the selected centers and $\mu_{q_k} = \begin{bmatrix}\mu_{q_k}^x \\ \mu_{q_k}^y\end{bmatrix}$.

In a second phase, the proposed method determines a linear transformation for each neighborhood using a least squares criterion. Local modeling favors simple models and a simple training criterion. The linear regression model is:

$$\left(\mu_{q_k}^x\right)^T \cdot W = \left(\mu_{q_k}^y\right)^T \tag{6.9}$$

The linear transformation W is obtained by solving:

$$N^x \cdot W = N^y \tag{6.10}$$

which has the least squares solution:

$$W = ((N^x)^T N^x)^{-1} (N^x)^T N^y \tag{6.11}$$

where $N^x = \left[\mu_{q_1}^x, \mu_{q_2}^x, \dots, \mu_{q_K}^x\right]^T$ and $N^y = \left[\mu_{q_1}^y, \mu_{q_2}^y, \dots, \mu_{q_K}^y\right]^T$.

The least squares solution minimizes the criterion:

$$C = \sum_{k=1}^{K} \left\| \left(\mu_{q_k}^{x}\right)^T \cdot W - \left(\mu_{q_k}^{y}\right)^T \right\|^2 \tag{6.12}$$

Finally, the converted result for x is computed as:

$$\left(y_{conv}\right)^T = x^T \cdot W \ . \tag{6.13}$$

The conversion of an entire sequence of source vectors can be obtained by repeating for each vector the procedure described above.

In practice it was noticed that the quality of the conversion is sensitive to the selected neighborhood and the type of linear transformation used. Firstly, it was found beneficial to estimate band diagonal matrices instead of full ones given that the correlation is highest between neighbor elements of an LSF vector. Secondly, it was found beneficial to use y_{conv} for a new selection of neighbors minimizing:

$$d_E\left(\begin{bmatrix} x \\ y_{conv} \end{bmatrix}, \mu_q\right) = \left\| \begin{bmatrix} x \\ y_{conv} \end{bmatrix} - \mu_q \right\| \tag{6.14}$$

(where $d_E(.,.)$ denotes the Euclidean or 2-norm distance between two vectors of the same dimension) and iterate the same steps until the neighborhoods determined in consecutive steps become virtually identical or sufficiently similar. This is equivalent to a convergence of y_{conv}. The process was found to be pseudo-convergent and can be stopped with an arbitrary threshold criterion. Figure 6.2 illustrates this pseudo-convergence. Intuitively, it is natural to involve the estimate y_{conv} in the selection of the neighbors because the original vector quantization was performed on joint feature vectors, not only on the source side.

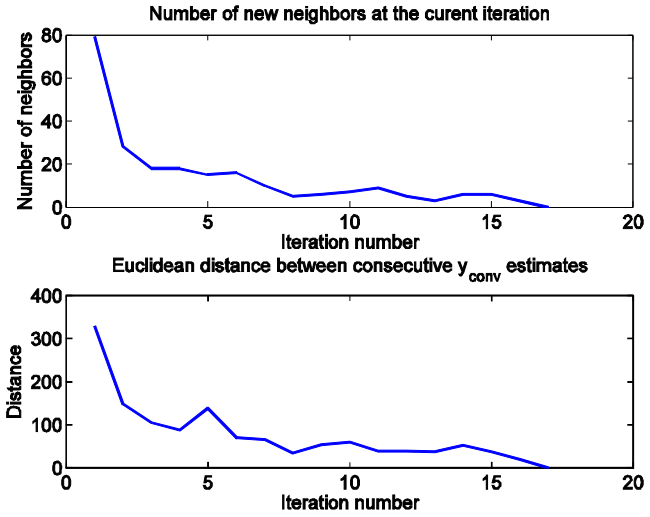

Figure 6.2: Pseudo-convergence of neighborhood selection. (from [48])

We observe that the algorithm could have been applied directly on the aligned training data instead of the codebook.

6.2.2 Experiments

The algorithm for spectral conversion presented in the previous section has been applied on 16-dimensional line spectral frequencies (LSF) vectors and the results are demonstrated in this section with two cross gender examples. The section presents a comparison with the popular GMM based approach providing objective and subjective results.

Acoustic data

CMU Arctic database [174] is a publicly available corpus of parallel speech sampled at 16 kHz. We used the CLB (female) and RMS (male) speakers from the CMU Arctic database to test conversion in both directions: from male voice to female and from female voice to male.

A parallel set of 100 sentences was used as training data amounting to approximately 30000 pairs of source and target LSF vectors after time alignment. Another 10 sentences were used for testing.

Model settings

GMM:

Too few components, although reliably estimated, give an inaccurate approximation of the training data while the estimation of too many components is unreliable causing over-fitting. In choosing a reference GMM for the comparison with the proposed approach such problems are avoided as follows. The performance of GMM models with different numbers of components was evaluated over the test set and the model with the lowest error was selected.

As illustrated in Figure 6.3 the female to male direction requires 8 components while 16 components are needed to convert the male into female voice. The mean squared error (MSE) figures are based on the definition given in [40] and reproduced in section 6.3.3.

Even though the GMM was tuned directly on the test set, a similar tuning could be performed by cross-validation using only the training set.

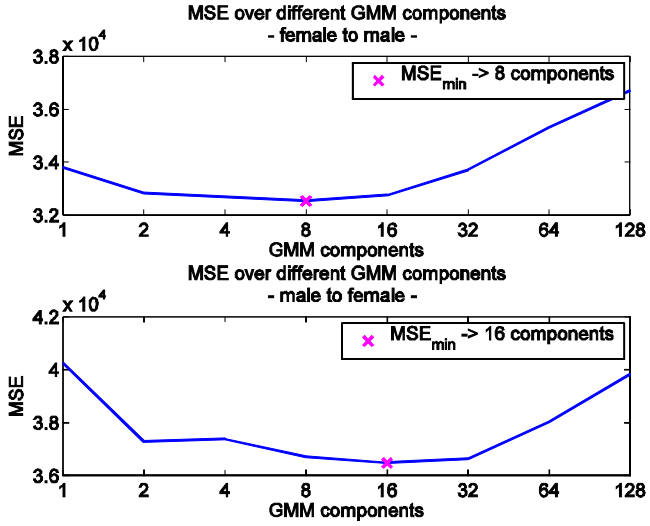

Figure 6.3: Mean squared error of GMMs with different numbers of components measured over the test set. (from [48])

Proposed method. (Local linear transformation):

The tuning of the proposed method is mainly based on perceptual evaluation. A codebook size of 8000 was used while the neighborhood sizes were tuned separately for each direction leading to values of 40 (female to male) and 130 (male to female). The linear transformations were restricted to tri-diagonal matrices.

The neighborhood size was found to act as a tradeoff producing unstable results when the neighborhood is too small and excessively averaged (over-smoothed) results when neighborhoods are large.

Subjective listening test

The speech samples evaluated in the listening tests are based on target speaker versions of the test utterances, in whose parametric representations only LSFs have been replaced with converted ones. This mimics the case when all other features are ideally converted focusing the evaluation on the actual spectral conversion.

For each conversion direction a modified MOS test was carried out by ten listeners on ten test sentences. The proposed method (LLT) and the GMM based approach were compared in terms of speech quality and success of identity mapping. These criteria are evaluated with scores between -2 and 2 with -2/+2 indicating that "GMM/LLT performs much better", -1/+1 for "GMM/LLT performs better" and 0 indicating perceptually identical performance. The results of the listening test are illustrated in Table 6.2.

Table 6.2: Subjective listening test scores with 95% confidence intervals.

	Quality	Identity
Female to male	0.49±0.19	0.33±0.17
Male to female	0.23±0.17	0.15±0.16

A possible explanation for the male to female result is that the high pitched female voice seems to mask the quality problems making the two methods sound more similar.

The subjective scores indicate the general preference of the proposed approach over the GMM based system.

Over-smoothing reduction

The converted spectra and LSF tracks illustrated in Figure 6.4 indicate a reduction of the over-smoothing in the case of the proposed approach in comparison to GMM.

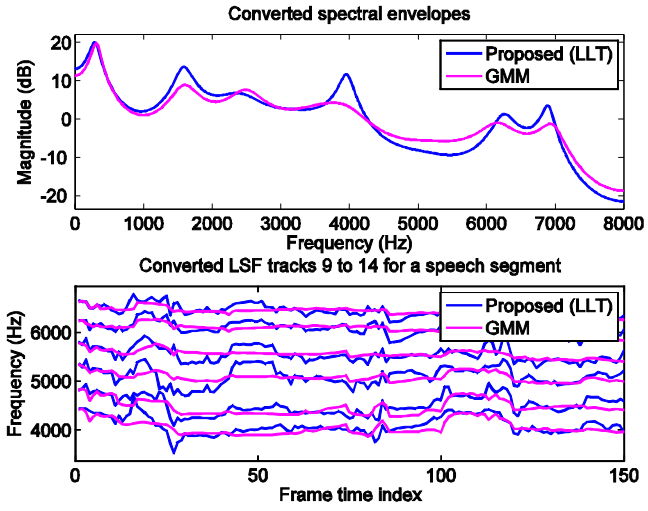

Figure 6.4: Over-smoothing reduction for spectral envelopes (top) and LSF tracks (bottom). (from [48])

Standard deviation measurements of converted and original target spectra (in frequency) and LSF tracks (in time) are calculated over the entire test set and summarized in Table 6.3 confirming the over-smoothing reduction.

Table 6.3: Average standard deviation of spectral magnitude (in dB) and LSF tracks (in Hz)

	Magnitude (dB)			LSF tracks (Hz)		
	Proposed	GMM	Tgt.	Proposed	GMM	Tgt.
Female to male	8.19	7.46	8.74	237	199	264
Male to female	10.70	9.86	10.65	336	296	328

Local modeling, rather than the interpolation of local models from acoustically remote regions, makes the proposed approach capable to capture details better and reduce the averaging effect.

6.2.3 Discussion

The method was shown to effectively reduce over-smoothing and obtained favorable preference scores in a subjective evaluation against the popular GMM based approach. On the downside the proposed method uses heavier computation for conversion as linear transformations depend on the input vector and have to be estimated at runtime.

In the proposed method the use of tri-diagonal matrices instead of full ones led to the improvement of the results. This means that each element $y(k)$ of a target LSF vector y is determined as a linear combination of the elements $x(k-1)$, $x(k)$ and $x(k+1)$ of an aligned source LSF x. This solution is based on the fact that the cross correlation of $y(k)$ is usually highest with these elements and it is sensible to try to predict $y(k)$ from highly correlated data. Therefore we only determine the band diagonal elements of the transformation matrix. However, it is not guaranteed that this choice represents accurately the highest correlations. For this reason, finding and determining the matrix elements that correspond to the highest correlations and using such a matrix instead of the proposed band-diagonal one might lead to further improvements of the results. In this case the matrix might have some generic sparse form since it leaves out the low correlation elements. Particularly for small data it is important to work with sparse matrices and avoid estimating unreliable matrix elements. Alternative regularization procedures can be considered for determining an optimal sparse matrix that describes most accurately the input-output relationship.

Interesting directions for future work would be to study alternative ways for neighborhood selection and alternative local models.

6.3 BILINEAR MODELS

This section presents a voice conversion technique based on bilinear models and introduces the concept of contextual modeling, both published in [40]. The bilinear approach reformulates the spectral envelope representation from line spectral frequencies feature to a two-factor parameterization corresponding to speaker identity and phonetic information, the so-called style and content factors. This decomposition offers a flexible representation suitable for voice conversion and facilitates the use of efficient training algorithms based on singular value decomposition. In a contextual approach (bilinear) models are trained on subsets of the training data selected on the fly at conversion time depending on the characteristics of the feature vector to be converted. The performance of bilinear models and context modeling is evaluated in objective and perceptual tests by comparison with the popular GMM-based voice conversion method for several sizes and different types of training data.

The bilinear models represent a factor analysis technique introduced originally in [175] which attempts to model observations as a result of two underlying factors. This concept originated from the observation that living organisms are capable of separating "style" and "content" in their perception. The separation into these two factors gives a flexible representation and facilitates the generalization to unseen styles or content classes. Furthermore, this framework provides efficient training algorithms based on singular value decomposition (SVD). In [47] we have demonstrated with early results that bilinear models are a viable solution also for voice conversion, by studying the voice conversion in terms of style (speaker identity) and content (the phonetic information) with small parallel sets of training data.

Due to their capability for reconstructing missing data, we hypothesize that bilinear models may be particularly useful in text independent cases and especially in cross-lingual voice conversion. This assumption will be evaluated using the alignment scheme for text-independent data. The proposed conversion technique based on bilinear models is compared with the widely used GMM based method using both parallel and text independent data with very small to very large sizes of the training sets. Our results offer a comprehensive perspective over the performance and the limitations that bilinear models have in voice conversion. In addition, we also try to answer the question whether fitting conversion models to contextual data (a subset of the training data) is more appropriate for capturing details than the usual models optimized globally over the entire training data.

6.3.1 Voice Conversion with Asymmetric Bilinear Models

The general style and content framework originally presented in [175] can be successfully utilized for spectral transformation in voice conversion. This section describes the asymmetric bilinear models following the notations used in [175], and discusses the properties of the technique from the voice conversion perspective. In the following, we will use the terms *style* and *content* to refer to the *speaker identity* and *phonetic information*, respectively, which constitute the two independent factors underlying our observations. In this section, the observations are represented as line spectral frequency (LSF) vectors.

Asymmetric bilinear models

In a symmetric model, the style s (the speaker identity) and content c (the phonetic information) are represented as parameter vectors denoted a^s and b^c of dimension I and J, respectively. Let y^{sc} denote an observation vector in style s and content class c, and let K denote its dimension. In our case, y^{sc} is an LSF vector of one speaker and it represents the spectral envelope of a particular speech frame. y^{sc} as a bilinear function of a^s and b^c, in its most general form, is given by [175]

$$y_k^{sc} = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ijk} a_i^s b_j^c \tag{6.15}$$

where i, j and k denote elements of the style, content and observation vectors. The terms w_{ijk} describe the interaction between the content (phonetic information) and style (speaker identity) factors and are independent of both of these factors.

Asymmetric bilinear models are derived from the symmetric bilinear models by allowing the interaction terms w_{ijk} to vary with the style leading to a more flexible style description [175]. Equation (6.15) becomes

$$y_k^{sc} = \sum_{i,j} w_{ijk}^s a_i^s b_j^c \ . \tag{6.16}$$

Combining the style(identity)-specific terms in equation (6.16) into

$$a_{jk}^s = \sum_i w_{ijk}^s a_i^s \ , \tag{6.17}$$

gives

$$y_k^{sc} = \sum_j a_{jk}^s b_j^c \ . \tag{6.18}$$

By denoting as A^s the K x J matrix with entries a_{jk}^s, equation (6.18) can be rewritten as

$$y^{sc} = A^s b^c \ . \tag{6.19}$$

In this formulation the a_{jk}^s terms can be interpreted as a style (identity) specific linear map from the content (phonetic info) space to the observation space (LSF). It is worth to note that unlike the face image case presented in [175] in which basis vectors appeared to have some concrete interpretations, no obvious patterns could be observed and no meaningful interpretation could be attributed to the parameter vectors and matrices in our particular application.

Model fitting procedure

The objective of the model fitting procedure is to train the parameters of the asymmetric model to minimize the total squared error over the entire training dataset. This is equivalent to maximum likelihood (ML) [123] estimation of the style and content parameters based on the training data, with the assumption that the data was produced by the models plus independently and identically distributed (i.i.d.) Gaussian noise [175].

The model fitting is described for S speakers (styles) and C content classes which could correspond to phonetically justified units. Our training material consists of R_1 LSF vectors of speech uttered by speaker s_1 in style $s=1$, R_2 LSF vectors of speech uttered by speaker s_2 in style $s=2$, and so on. The individual (speaker based) parametric sequences are pooled together in a training sequence of size $R = R_1 + R_2 + \cdots + R_S$. Let $y(r)$ denote the r^{th} training observation ($r = 1, \ldots, R$) from the pooled data. Each $y(r)$ is an LSF vector coming from a certain speaker (style) and from one of C content classes. The binary indicator variable $h^{sc}(r)$ takes the value *1* if $y(r)$ is in style s and content class c and the value *0*, otherwise. The total squared error E of the asymmetric model given in equation (6.19) is computed over the training set using

$$E = \sum_{r=1}^{R} \sum_{s=1}^{S} \sum_{c=1}^{C} h^{sc}(r) \| y(r) - A^s b^c \|^2 \ . \tag{6.20}$$

In the case of parallel training data, the speech sequences of the S speakers can be time aligned and each S-tuple of aligned LSF vectors will be assumed to represent a distinct class. Consequently, there will be only one LSF vector from each speaker (style) falling into each content class.

If the training set contains an equal number of observations in each style and in each content class (in our case one observation), a closed form procedure exists for fitting the asymmetric model using singular value decomposition (SVD) [175].

In the proposed case of parallel and aligned training data, in order to work with standard matrix algorithms, we stack the SC *(=R)* LSFs (K dimensional column) vectors into a single SK x C matrix, similarly as in [175],

$$Y = \begin{bmatrix} y^{11} & \dots & y^{1C} \\ \dots & \dots & \dots \\ y^{S1} & \dots & y^{SC} \end{bmatrix}. \quad (6.21)$$

We can express now the asymmetric model in the following very compact matrix form

$$Y = AB, \quad (6.22)$$

where the (SK)xJ matrix A and the JxC matrix B represent the stacked style and content parameters,

$$A = \begin{bmatrix} A^1 \\ \dots \\ A^S \end{bmatrix}, \quad (6.23)$$

$$B = [b^1 \quad \dots \quad b^C]. \quad (6.24)$$

To find the optimal style and content parameters for equation (6.22) in the least square sense, we can compute the SVD of $Y = UZV^T$ [175] with complexity $O(min((SK)^2C,(SK)C^2))$. (Z is considered to have the diagonal eigenvalues in decreasing order.) By definition, we choose the style parameter matrix A to be the first J columns of UZ and the content parameter matrix B to be the first J rows of V^T. There are many ways to choose the model dimensionality J e.g. from prior knowledge, by requiring a desired level of approximation of data, or by identifying an "elbow" in the singular value spectrum [175].

Note that using a relatively small model order $J = S \cdot K$ prevents overfitting and that potential numerical problems due to very large matrices can be avoided by computing an *economy size* decomposition (in Matlab).

An important aspect in cases with very high dimensional features is the selection of the model dimensionality (J) since high model dimension could cause over-fitting. Our experiments with $K=16$ and $K=10$ dimensional LSFs produced similar results with the difference that the error decreases and stabilizes faster for K=10 as fewer parameters require less data for a reliable training.

Application in parallel voice conversion

One of the tasks that fall under the framework proposed in [175] and which is of particular interest in voice conversion is *extrapolation* illustrated in Table 6.4. In this character example the letters D and E (content classes) do not exist and need to be generated in the new font (style) based on the labeled training set (first two rows) [175].

Table 6.4: The extrapolation task illustrated for characters

A	B	C	D	E
A	*B*	*C*	*D*	*E*
A	B	C	?	?

The term extrapolation refers to the ability to produce equivalent content in a new style, in our case to produce speech as that uttered by a source speaker but with a target speaker's voice. Therefore, voice conversion is a direct analogy of the extrapolation task. Extending a bit the concept of voice conversion we can also define it as the generation of speech with a target voice, reproducing content uttered by multiple source speakers.

We can formulate the problem of parallel voice conversion as an extrapolation task as follows. Given a training set of parallel speech data from S source speakers and the target speaker, the task is to generate any test sentence in the target voice starting from S utterances of the test sentence corresponding to each of the S source voices (styles).

The alignment of the training data (S source + one target speakers) is a prerequisite step for model estimation and is usually done with DTW. On the other hand, the alignment of the test data (S utterances of the source speakers) is also required if $S>1$. The test data is aligned to a target utterance of the test sentence which exists in this study for evaluative purposes. In real applications, where such a target utterance does not exist, the test data should be aligned to one of its S source utterances, preferably a source speaker (denoted as main source speaker) whose speaking style resembles that of the target speaker. Choosing the alignment in this way has at least two advantages: provides a natural speaking style for the converted utterance which is close to the target one and reduces alignment problems because at least the main source speaker's utterance does not have to be interpolated in the alignment process.

A so-called *complete* data is formed by concatenating the aligned training and test data of the S source speakers. The *complete* data is assumed to have as many classes as LSF vectors per speaker and is used to fit the asymmetric bilinear model of equation (6.22) to the S source styles following the closed-form SVD procedure described previously in the section *"Model Fitting Procedure"*. This yields a K x J matrix A^s for each source style (voice) s and a J dimensional vector b^c for each LSF class c in the *complete* data (hence producing also the b^c-s of the test utterance).

The model adaptation to the incomplete new style t (the target voice) can be done in closed form using the content vectors b^c learned during training. Suppose the aligned training data from our target speaker (style t) consists of M LSF vectors which by convention we considered to be in M different content classes $C_T = \{c_1, c_2, \dots, c_M\}$. We can derive the style matrix A^t that minimizes the total squared error over the target training data,

$$E^* = \sum_{c \in C_T} \| y^{tc} - A^t b^c \|^2 . \tag{6.25}$$

The minimum of E^* is found by solving the linear system

$$\frac{\partial E^*}{\partial A^t} = 0 . \tag{6.26}$$

The missing observations (LSFs) in the style t and a content class c of the test sentence can be synthesized from $y^{tc} = A^t b^c$. This means we can estimate the target version of the test sentence by multiplying the target style matrix A^t with the content vectors corresponding to the test sentence.

The proposed algorithm

The proposed technique is summarized in the following algorithm in which we assume that LSF features are available.

1) Time align the training data (source speakers and target speaker) and the test sentence (source speakers only) which is to be converted to the target voice. The alignment will respect the timeline or prosody of the main source speaker.
2) Form the *complete* data of the source speakers by combining their training data with their test sentence data.
3) Run SVD to fit the asymmetric bilinear model to the *complete* data. This step will find the style matrices A^s for all the source speakers and the content vectors b^c for all the content (LSF) classes, including the classes (LSFs) in the test sentence.
4) Find the style matrix A^t of the target voice by minimizing the criterion given in equation (6.25), thus solving equation (6.26).

5) Synthesize the converted LSF vectors as $y^{tc} = A^t b^c$ with A^t found at step 4 and the content vectors b^c of the test sentence found at step 3.

The non-parallel case

Due to their capability for reconstructing missing data, we hypothesize that bilinear models may be particularly useful in text independent cases and especially in cross-lingual voice conversion.

In order to evaluate and compare the performances of bilinear and GMM models with text independent data we have extracted one non-parallel and one artificially cross-lingual subset from the original parallel intra-lingual corpus. The so-called simulated cross-lingual corpus was designed by ensuring that the target speaker utilizes in his utterances only a subset of the phonemes used by the source speakers.

By observing that the correspondence of speech content between speakers has moved from the utterance level to the phoneme level we use the alignment scheme based on temporal decomposition and phonetic segmentation of the speech signal presented in section 4.2 inspired from [16]. The role of the alignment scheme here is to facilitate the use of parallel voice conversion algorithms with text independent data allowing us to focus on the evaluation of GMM and bilinear models.

6.3.2 Contextual Modeling

The traditional GMM based voice conversion methods fit a GMM to the aligned training data globally without any explicit consideration of the various phoneme classes. It is natural to question whether GMM is able to capture the fine details of each phonetic class when the training optimizes a global fitting. It is also natural to wonder whether these details are influenced or not by the local context. It is not practical though to train a different model for each different context or even for each different phoneme due to the large amount of data necessary for such training. The research conducted so far has not been able to give clear answers to these questions.

To shed some light on the above issues more closely, we have studied the use of contextual modeling in voice conversion. By contextual modeling we refer to a scheme in which multiple models are optimized on possibly overlapping subsets of the training data denoted as contexts. We hypothesize that such a modeling could potentially offer more accuracy and partially alleviate the known over-smoothing problem of the traditional GMM based techniques.

Each feature vector y_i in the parameterized speech sequence $[y_1, y_2, \dots, y_N]$ can be regarded as belonging to a context and is associated with a context descriptor α_i. For simplicity α_i can be regarded as the phonetic unit to which y_i belongs but in a broader sense the context descriptor can be any meaningful parameter (e.g. dy/dt, the time derivative of y).

For the conversion of a feature vector y_i we first select the appropriate conversion model based on its context descriptor α_i. A potentially different model is selected for the conversion of a different feature vector y_j.

Since it is not practical to train and store models for thousands of contexts beforehand, we can perform model training on context data selected on the fly for each feature vector y_i based on α_i.

Context data may be considerably small depending on the selection rule (it is not practical to gather sufficient data to train e.g. a reliable phoneme model) therefore the trained models need to be robust with small data, fast and computationally efficient because they are trained repeatedly on different contexts. Our results presented in [47] recommend bilinear models for this task.

Practical implementation

In order to demonstrate the concept of contextual modeling we consider the same scenario with multiple source speakers and one target speaker whose training data is aligned with the technique presented in section 4.2 in the form of aligned event target vectors. The proposed algorithm requires aligned event target representations of the test utterance from all the source speakers. Furthermore we use phonetic annotations in order to segment the aligned representations into phonetic units as defined in section 4.2. Blocks of SxQ_{pu} event targets representing one phonetic unit are converted one at a time generating Q_{pu} converted event targets as suggested in Figure 6.5 below, with S being the number of source speakers and Q_{pu}=4 the number of equally spaced event targets used to represent one phonetic unit.

Let $\theta_g, \theta_f \in \Theta$ be the $(j\text{-}1)^{\text{th}}$ and j^{th} phonemes of the test utterance, alternatively denoted as p_{j-1} and p_j respectively. We note that each phonetic unit (e.g. $p_{j-1}p_j = \theta_g\theta_f$) corresponds to a node of the matrix D (e.g. (g,f)) representing the full training data as introduced in section 4.2.

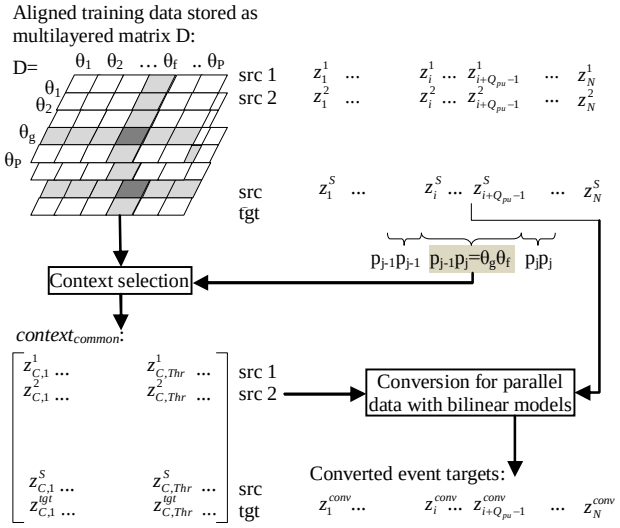

Figure 6.5: Context selection for the current phonetic unit and the conversion of its corresponding block of event vectors. (from [40])

For each phonetic unit of the test utterance a context data $context_{common}$ is extracted from the full training set using the multilayer matrix structure D. To illustrate the selection we describe next how this is done for the phonetic unit $p_{j-1}p_j = \theta_g\theta_f$.

1) Start with an empty $context$ data. $context = \varnothing$.
2) Add the data corresponding to the current phonetic unit ($p_{j-1}p_j = \theta_g\theta_f$). $context = context \cup D(g,f)$.
3) If $size(context_{common}) < Thr$ then $context = context \cup D(k,f)$, $1 \le k \le P$, $k \ne f$ and $context = context \cup D(g,k)$, $1 \le k \le P$, $k \ne g$. By $common$ data we refer to any $D(l,m)$ for which both θ_l and θ_m represent phonemes common to both source and target speakers, $context_{common}$ represents the common part of the data included in the current $context$, and Thr denotes a size threshold.

4) For $p_{j-1}p_{j-1}$, p_jp_j, $p_{j-2}p_{j-1}$, p_jp_{j+1}, $p_{j-2}p_{j-2}$, $p_{j+1}p_{j+1}$, $p_{j-3}p_{j-2}$, $p_{j+1}p_{j+2}$ … until $size(context_{common}) \geq Thr$, do step 2 and step 3 (if such a unit is within the utterance bounds), but in the context building skip the nodes of D that have already been collected.

By construction $context_{common}$ is an aligned dataset of event targets of all the S source speakers and the target speaker. The block of source event targets corresponding to the phonetic unit for which $context_{common}$ was built can be converted using this context data and the bilinear models framework for parallel data from section 6.3.1.

After the conversion of the event targets the desired number of feature vectors can be reconstructed using event functions.

6.3.3 Experiments and Results

This work extends the study of voice conversion with bilinear models from the case of parallel and limited training sets [47] to non-parallel and simulated cross-lingual cases evaluating how the size of the training data and the contextual modeling influence the performance. Unlike in [47], the bilinear model is now compared against a GMM whose number of mixture components is optimized for the amount of available training data. Both objective metrics and listening test results are used. The GMM is chosen as a reference because it has been well studied and its performance level should be familiar in the field of voice conversion.

The experimental set-up

The present study is concerned only with the spectral conversion and does not discuss prosodic nor energy conversion. We use 16-dimensional LSF vectors for the representation of the spectral envelope as proposed in [36]. LSFs relate closely to formant frequencies but unlike formant frequencies they can be reliably estimated [176][177]. They have also favorable interpolation properties and local spectral sensitivity which means that a badly estimated component affects only a small portion of the spectrum around that frequency [178][179]. Interestingly, LSFs have also been used with MRTD due to these beneficial characteristics.

We used in our experiments two source speakers (male and female) and one target speaker (male) selected out of four US English speakers available in the CMU Arctic database. The Arctic database is a parallel corpus of 16 kHz speech samples provided with phonetic labels and it is publicly available [174]. The samples consist of short utterances with an average duration of 3 seconds.

The number of three speakers is not meant to be an optimal lower limit, they were chosen with the purpose of ensuring sufficiently large and phonetically balanced text independent partitions of this parallel database. Another criterion was to have an equal number of male and female source speakers. It is assumed [168] that an increased number of speakers would be beneficial for the proposed bilinear method leading to a better separation of the style and content factors.

Phonetically balanced sets of utterances were selected from each speaker to form parallel, non-parallel and simulated cross-lingual training corpora with 3, 10, 70, 140 and 264 utterances.

In parallel and non-parallel training data all speakers cover the full US English phoneme set but in the case of simulated cross-lingual data only the two source speakers use the full phoneme set. We simulate a cross-lingual corpus by defining a set of 5 *rare* phonemes and selecting in the training data only target utterances in which none of the *rare* phonemes occurs. The benefit from doing so is that, unlike in the real cross-lingual case, we can evaluate the conversion of phonemes *unseen* in the target

training data against real target instances of these *rare* phonemes. The selected *rare* phonemes are those with the lowest rate of occurrence in the database: 'zh' as in "mirage"/m-er-aa-zh, 'oy' as in "joy"/jh-oy, 'uh' as in "could"/k-uh-d, 'ch' as in "charge"/ch-aa-r-jh, 'th' as in "author"/ao-th-er. This selection attempts to make efficient use of the full parallel data by maximizing the size of its cross-lingual partition and does not guarantee a minimal acoustic similarity between the *rare* phonemes and other *common* phonemes used in training. The resemblance is possible to some extent (e.g. 'th'≈'t') but seems to be rather limited. In our study the transcriptions are assumed to be accurate and no special handling is provided for pronunciation differences.

The alignment of the LSF vectors from parallel data is accomplished using dynamic time warping (DTW) on Mel-frequency cepstral coefficients (MFCC) extracted at the same time locations as the LSFs. For non-parallel and simulated cross-lingual data, event target vectors are aligned following the procedure described in section 4.2.

The bilinear method presented in section 6.3.1 and the contextual modeling method described in section 6.3.2 are compared against a modified GMM based method. The modified GMM method uses data from two source speakers to predict the target speaker's voice in the same way as the original GMM method uses data from one source speaker to predict the target voice. Our tests indicate that the modified approach outperforms the original model in terms of mean squared error. The modified method requires aligned data from the three speakers to train a conversion model whose input is a concatenation of two aligned feature vectors from the two source speeches and whose output is a feature vector of the target speech. With the above specification the GMM training and conversion are done as described in [124]. It is worth to observe that in the simulated cross-lingual case only the *common* phonemes are represented in the data used to train the GMM. To keep comparisons between GMMs meaningful, the initialization of the GMM training is done always from the same list of data points in the same order. This way two GMM's with the same number of mixtures trained on different datasets would still be initialized identically.

To simplify the alignment in the test set, but also for a more meaningful evaluation of the conversion result, we design the test set as a phonetically balanced set of ten parallel utterances covering the entire phoneme set (including the *rare* phonemes). Including the rare phonemes is important especially for the evaluation of the simulated cross-lingual voice conversion.

Even though in real applications the test sentence does not exist in the target voice, in our study such an utterance exists and is used to align the test utterances of the source speakers to the speaking rate of the target speaker. This facilitates distance measurements in the feature domain between the converted LSF and real target LSF vectors and allows the converted LSFs to be used along with the rest of the original target parameters for the synthesis of a converted waveform. Hence the converted waveforms mimic the case when all other features except LSFs are ideally converted allowing the evaluation to more effectively focus on the performance of the spectral LSF conversion.

The contextual model experiment is run only once for the largest cross-lingual dataset (264 utterances) which is believed to ensure sufficient data for the training contexts. The conversion is done one phonetic unit at a time and for every phonetic unit a context is built by requiring at least 1000 aligned *common* frames (event targets). The size of 1000 was selected based on preliminary experiments and corresponds usually to 2-3 neighboring phonetic units in the speech sequence (i.e. *offset=±1 or ±2*).

About 3h 40min of contextual training was needed for the conversion of a 3 sec test utterance with a simulated cross-lingual training set of 264 utterances. For the same data the typical time

required to train a GMM with 8 mixtures is 2 min while a bilinear model or a GMM with 1 mixture takes about 2 sec to train. The times are reported for an Intel Core2 CPU 6300 @ 1.86 GHz with 1Gb of memory.

Metrics for objective evaluation

The first objective metric used is the *mean squared error* (MSE) which is computed between a converted and a target LSF vector using the formula

$$MSE(lsf_c, lsf_t) = \frac{\sum_{i=1}^{N}(lsf_c(i) - lsf_t(i))^2}{N}, \tag{6.27}$$

where lsf_c and lsf_t denote the converted and target LSF vectors and N represents the LSF order. The frame-wise MSE figures are then averaged over the entire test data.

Spectral distortion (SD) is computed between a converted spectral envelope (derived from the converted LSF) and the corresponding target spectral envelope. The SD is measured only for a selected frequency range of the spectrum, using

$$SD^2 = \frac{1}{(f_u - f_l)} \int_{f_l}^{f_u} \left(20log_{10} \frac{\left|H(e^{j2\pi f/f_s})\right|}{\left|\widehat{H}(e^{j2\pi f/f_s})\right|} \right)^2 df \tag{6.28}$$

where H and $\widehat{H}$ represent the target and converted spectra, respectively, f_s is the sampling frequency, and f_l and f_u denote the frequency limits of the integration. For better perceptual relevance, SD is computed between 0 and 4 kHz.

The relationship between the training data size and the number of components for GMM

We studied with parallel training data how the GMM performance is related to the number of mixtures and the size of the training set.

For reduced datasets (3 utterances) the best GMM performance is attained using one mixture component. Objective results in Figure 6.6 and indeed perceptual ones presented later in this section, indicate a close tie between this configuration and the bilinear approach.

On the other hand, four mixtures achieve optimal or close to optimal performance for larger sets (70-264 utterances). With 264 utterances for instance, a degradation to a lesser or larger extent is produced for less than 4 or more than 8 mixture components.

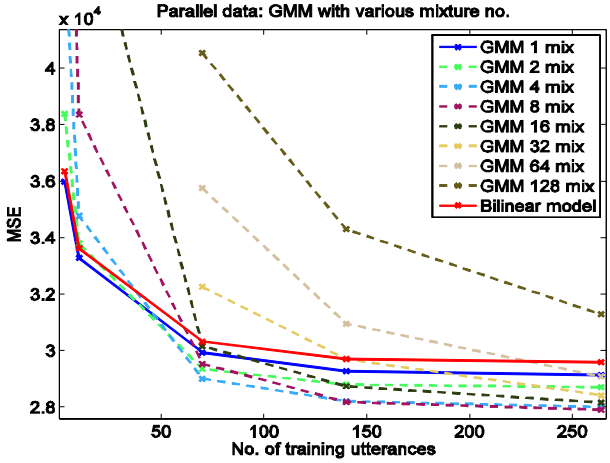

Figure 6.6: MSE for GMMs with different mixture numbers and training sizes demonstrated with parallel training data; a similar figure for the bilinear approach is superimposed. (from [40])

It is difficult to know beforehand what number of mixture components is optimal for a given amount of training data. Too few components, although reliably estimated, would give an inaccurate approximation of the training data while estimating too many components becomes unreliable and may cause over-fitting problems.

The result obtained with bilinear models was superimposed in Figure 6.6 for comparison revealing an interesting similarity with the one mixture case of the GMM. Both models outperform all other GMM configurations for small training sets but remain slightly behind for large data. It is worth to notice that the proposed bilinear model does not require preliminary order tuning.

The result presented in this section was used to determine an "optimal" number of mixture components for the GMMs involved in the next sections depending on the amount of aligned LSF vectors in the training data.

Objective results

The objective results obtained for a training set of 3 parallel utterances are shown in Table 6.5. The "optimal" number of components for GMM in this case is 1.

Table 6.5: Mean squared error (MSE) and spectral distortion (SD) results for 3 parallel training utterances

	Bilinear model	GMM (1 mix)
MSE	36625	36329
SD (dB)	5.51	5.50

It is worth observing that these figures are extremely close indicating that the bilinear model achieves close to optimal performance for small datasets having the advantage that it does not require tuning.

Figure 6.7 presents MSE for both GMM and bilinear methods for parallel, non-parallel and simulated cross-lingual cases and for various sizes of the training data. The contextual modeling was evaluated only for 264 simulated cross-lingual utterances.

For GMMs, the "optimal" mixture numbers corresponding to 3, 10, 70, 140, 264 utterances were found to be 1, 1, 4, 8, 8 for parallel data and 1, 1, 4, 4, 8 for non-parallel and cross-lingual data.

The two techniques compare to each other similarly in all three scenarios. Their objective performance is very similar for small training sets while the "optimal" GMM gains some advantage for larger training sets. It is important to observe that this performance gain for the GMM is obtained at the cost of increased computational complexity corresponding to a larger number of mixture components. As reflected in the listening tests presented later this difference in objective measurements seems to be very small from a perceptual point of view.

We can also see that the contextual modeling brings a sensible improvement compared to the "optimal" GMM and bilinear models fitted globally on full cross-lingual training data (264 utterances).

Figure 6.8 shows the corresponding spectral distortion results. An interesting aspect to note is that the minimum spectral distortion with 264 utterances is attained for the GMM method with parallel data (4.79 dB) while the maximum is 5.50 dB recorded for the bilinear approach with non-parallel data. The gap of only 0.71 dB is perceptually small and this was also reflected in the listening tests.

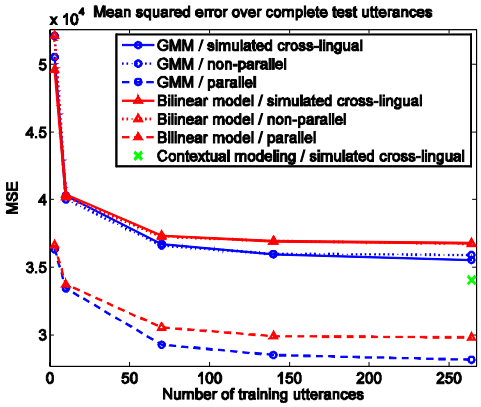

Figure 6.7:Mean squared error results over the set of test utterances. (from [40])

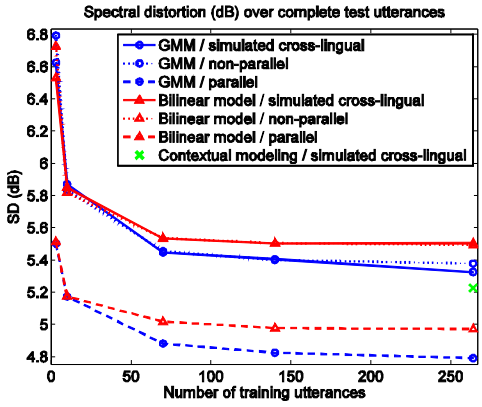

Figure 6.8: Spectral distortion (dB) results over the set of test utterances. (from [40])

Figure 6.9 and Figure 6.10 present consistent MSE and SD results for the conversion of the *rare* / *unseen* phonemes. It is interesting to observe in the simulated cross-lingual experiment the capability to restore phonemes *unseen* in the training data (*rare* phonemes). In the bottom plots of Figure 6.9 and Figure 6.10, we observe that the bilinear approach and the GMM based method perform similarly independent of the size of training data. By comparison with the cross-lingual results over complete utterances presented in Figure 6.7 and Figure 6.8, it is worth noticing that the error over the *unseen* phonemes is significantly larger.

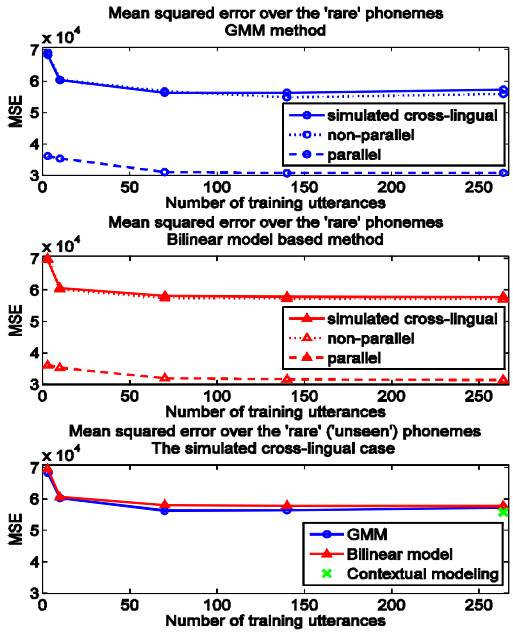

Figure 6.9: Mean squared error results over the *rare / unseen* phonemes existent in the test utterances. (from [40])

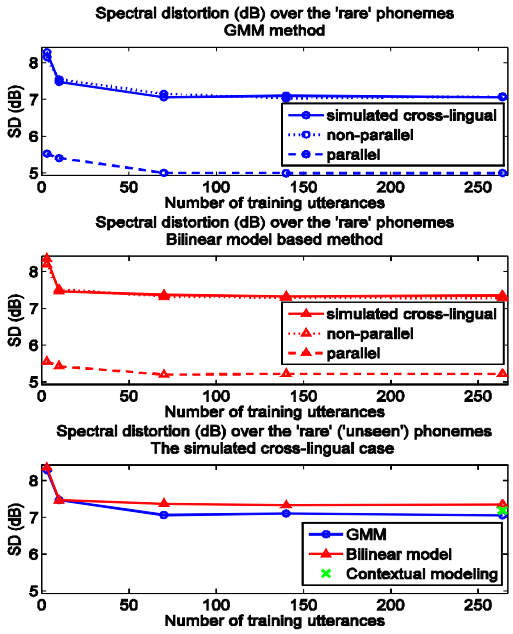

Figure 6.10: Spectral distortion (dB) results over the *rare / unseen* phonemes existent in the test utterances. (from [40])

The top and middle plots indicate for GMM and the bilinear method respectively that the accuracy of reconstruction is not depending much on whether the phoneme exists or not in the training data of the target speaker (minor differences between the results with non-parallel data including *rare*

phonemes and those with simulated cross-lingual data lacking them) but rather on the alignment and type of data (parallel or text independent). Better reconstruction results are obtained with parallel data which is also an indicator of the best performance that could be achieved due to its precise alignment and because the *rare* phonemes are included in the training. By comparison with the results in Figure 6.7 and Figure 6.8 we notice that the gap between figures for *rare* phonemes and complete utterances is significantly smaller for the parallel case than it is for the nonparallel and cross-lingual cases.

Interestingly, the result of the contextual modeling for the reconstruction of *unseen* phonemes is very similar to those obtained for the globally optimized GMM or bilinear models. This result is surprising considering that the missing phonemes are reconstructed based only on very small contexts of phonetic units. The bilinear model seems to be capable to generalize from a reduced subset of training data almost as well as it does when using the full data for training.

Figure 6.11 and Figure 6.12 illustrate a direct comparison between the two methods for every conversion scenario separately by showing again MSE and SD results measured over the entire test set.

Independent of the scenario, the performance of the bilinear models is very similar to that of the "optimal" GMM particularly for small training sets. While the objective results show a small performance advantage of the GMM for larger training sets, the subjective listening results presented next in this section indicate that the methods are still very close perceptually even for large datasets.

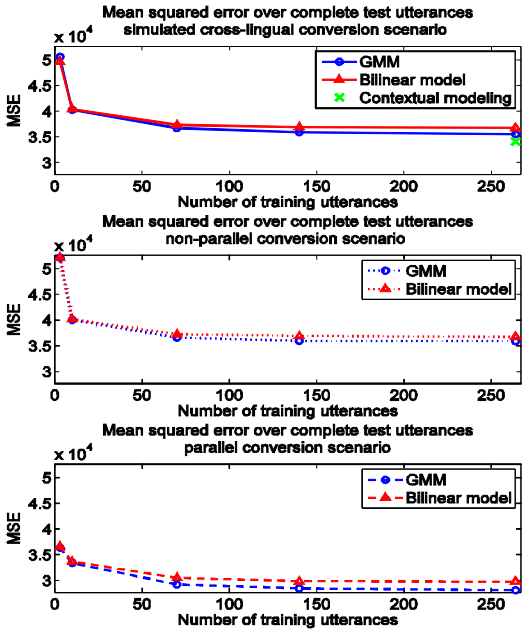

Figure 6.11: Comparative mean squared error results for the GMM, bilinear approach and contextual modeling in different conversion scenarios. (from [40])

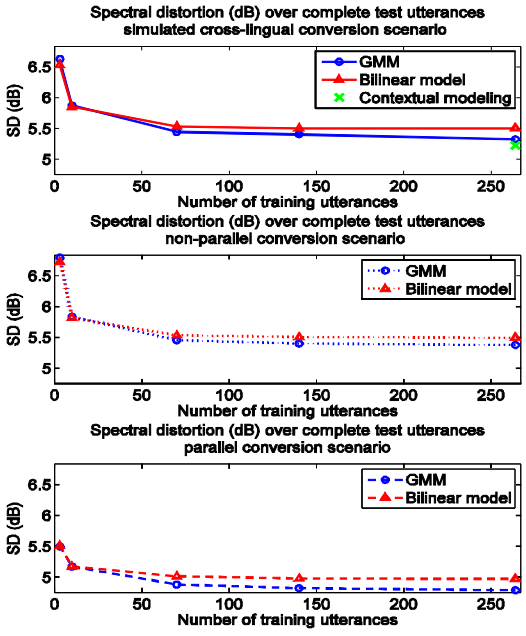

Figure 6.12: Comparative spectral distortion (dB) results for the GMM, bilinear approach and contextual modeling in different conversion scenarios (from [40])

The relatively small performance difference can be explained by observing the similarity in the MSE criteria that both methods optimize. The bilinear models optimize the criterion in equation (6.20) whereas the GMM optimizes a similar mean squared error criterion between the converted and target feature vectors. An interesting finding visible in the top panel reveals that contextual modeling

slightly outperforms the two techniques based on globally optimized models. This confirms that contextual approach may indeed capture details better than globally optimized models even though the gain does not justify the additional computational effort.

Finally, for each method (GMM and bilinear) we compared between three conversion scenarios: parallel, non-parallel and simulated cross-lingual (in Figure 6.13 and Figure 6.14 in terms of MSE and SD, respectively).

First, we notice that each method taken separately obtained very similar results for the non-parallel and simulated cross-lingual scenarios. This finding, in line with a similar result in the recovery of *rare* / *unseen* phonemes, indicates that the presence (in non-parallel data) or absence (from cross-lingual data) of the *rare* phonemes did not have a major influence on the results.

Secondly, both GMM and bilinear approaches perform clearly better with parallel training data than in the non-parallel or simulated cross-lingual cases and the difference is bigger for the small training sets (3 utterances).

The SD results shown in Figure 6.14 are again in line with the MSE scores presented above.

A concluding remark on the objective measurement experiments is that the performance of both systems is influenced by the amount of training data only up to a point beyond which adding more data does not bring significant improvements of the performance. We also note that it is the size of the actual aligned data that influences the performance and not the number of training sentences. A training data consisting of text independent utterances will result in significantly less aligned data than the same number of parallel utterances using our alignment technique. This also explains the bigger differences between parallel and text independent scenarios in the range 3 to 10 training utterances.

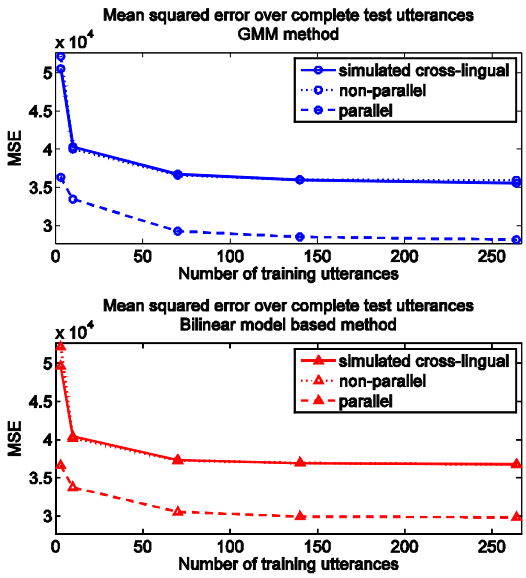

Figure 6.13: Comparative mean squared error results between different conversion scenarios for the GMM and bilinear approach. (from [40])

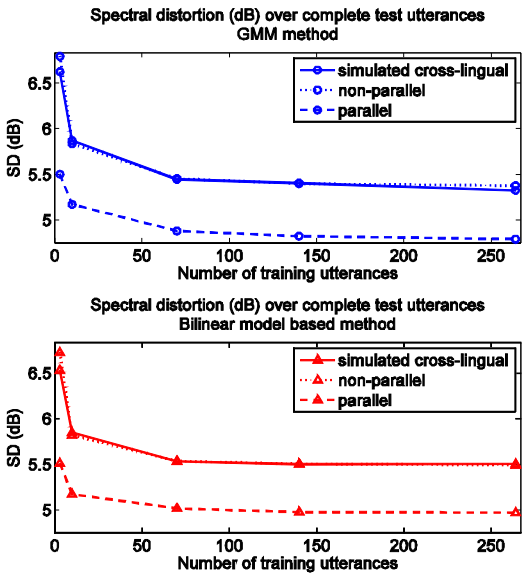

Figure 6.14: Comparative spectral distortion (dB) results between different conversion scenarios for the GMM and bilinear approach. (from [40])

Listening Tests

For a meaningful validation of the objective measurements we present subjective results with both reduced (3 utterances) and large training sets (264 utterances).

The first test compares the bilinear and GMM methods for a training set of 3 parallel utterances. One mixture component is used for the GMM as found "optimal" for the data size.

The next tests are concerned with large training sets of 264 utterances (approx. 3 sec per utterance) and use GMMs with 8 mixture components. They evaluate the GMM and bilinear methods relative to one another using parallel or simulated cross-lingual training data but also evaluate how these two scenarios (parallel and simulated cross-lingual) compare to each other for each of the two methods. In the last test the contextual modeling is compared with the GMM based method for the cross-lingual training data. The results with 95% confidence intervals are given in Table 6.6.

In each test, ten listeners compare schemes A and B using ten test utterances and a modified MOS test. In the identity test a real target version of the test sentence is compared in terms of voice identity with the converted samples obtained with schemes A and B. The quality test is simply a comparison in terms of speech quality between the two converted samples A and B. The successfulness of identity conversion and the overall speech quality are evaluated separately with scores between -2 (scheme A is much better than B) and 2 (scheme B is much better than A). The 0 (zero) score is given for perceptually identical performance.

The first result in Table 6.6 represents a comparison between the GMM based method with one mixture component and the bilinear approach for a training set of 3 parallel utterances. The very balanced score and its 95% confidence intervals indicate very similar performances for the two methods. This is in line with the objective results pointing out that the methods tend to have identical performance for small training sets. The SD figure shows a 0.01 dB difference (5.50 dB for GMM and 5.51 dB for the bilinear approach) which is not perceivable by humans.

Results for large training sets of 264 utterances are presented on lines 2 to 6 as follows. The second line compares the bilinear model and the 8-mixture GMM method found "optimal" for the given parallel data on which both methods are trained. The 95% confidence interval could not indicate a clear winner showing that the methods are perceptually equivalent. The SD difference of 0.17 dB (4.79 dB for GMM and 4.97 dB for bilinear) is hardly observable by the human hearing.

On the third line, the result obtained for the simulated cross-lingual scenario is slightly in favor of the GMM but the perceptual difference seems to be, however, very small. The exact 95% confidence interval actually extends by 0.0006 to the other side of the 0 axis, so in a strict sense it is impossible to call a winner.

Table 6.6: Subjective listening test results

Utts.	A	B	Quality	Identity
3	GMM/parallel	BL/parallel	0.02±0.08	0.01±0.07
264	GMM/parallel	BL/parallel	-0.08±0.12	-0.02±0.09
264	GMM/simulated cross-lingual	BL/simulated cross-lingual	-0.12±0.12	-0.05±0.09
264	GMM/parallel	GMM/simulated cross-lingual	-0.17±0.12	-0.11±0.10
264	BL/parallel	BL/simulated cross-lingual	-0.17±0.13	-0.01±0.10
264	GMM/simulated cross-lingual	CM/simulated cross-lingual	-0.05±0.12	-0.03±0.10

The fourth and fifth results of Table 6.6 indicate that both the GMM based method and the bilinear approach perform clearly better with parallel data than with simulated cross-lingual data but

interestingly the difference is very small. With -1 indicating that the parallel case is (clearly) better, and -2 for much better, our scores of ≈-0.17 could be interpreted as "only slightly better". This suggests that the type of the data may not be the essential factor for conversion as long as we have an efficient alignment scheme and that the proposed alignment scheme has been successful.

Finally, the last result of Table 6.6 represents a comparison between the "optimal" (8-mixture) GMM based method and the contextual modeling technique in the simulated cross-lingual case. Not surprisingly the 0.1 dB margin by which the context method outperforms the GMM approach is perceptually insignificant and the listening test result is consistent with this objective finding, indicating that it is practically impossible to decide a winner.

The listening test results are largely consistent with the objective measurements additionally revealing that the objective differences between the bilinear model and GMM, especially for large datasets, are very small or insignificant from a perceptual point of view. The small perceptual difference between parallel and cross-lingual scenarios is an indication of efficiency for our text-independent alignment. On the other hand the listening tests demonstrate that contextual modeling did not bring a perceptually meaningful gain.

6.3.4 Conclusions

This section presented a comprehensive study of bilinear models applied in voice transformation and explored their capability to reconstruct phonetic content in a new voice. The section also proposed a new conversion technique called contextual modeling that benefits from the efficient computation algorithms and the robust performance of the bilinear models with reduced data.

Objective and subjective evaluations of the bilinear model were reported in relationship to the traditional GMM-based technique with "optimal" number of mixture components determined based on the size of the training data. The objective figures of the two methods are particularly close in the range of small data while for larger sets the GMM seems to gain advantage. However, the listening tests indicated that the two methods perform equivalently or comparable from a perceptual point of view for both small and large training sets.

The gain in objective performance of the GMM for large data is achieved at the cost of an important increase of the computational complexity due to a larger number of mixture components. It is worth noticing that the bilinear model does not need any tuning.

Section 6.3.3 and [47] suggest that the bilinear model may have an important advantage in the range of small datasets over GMMs with more than one mixture component, both objectively and subjectively. This is demonstrated in [47] for a GMM with 4 mixtures. Section 6.3.3 also reflects with objective figures an interesting similarity between the bilinear model and a GMM with only one mixture.

The reconstruction capability of *unseen* data appears to be similar for the two methods independently of the data size.

Both in a global evaluation over the entire test set and exclusively over the *rare* phonetic units the non-parallel result is very similar to the simulated cross-lingual result, leading to an interesting finding that the performance is not influenced or influenced only marginally by incomplete data if sufficient training data is provided and sufficient *common* phonetic units are represented in the training data.

The performance seems to be much more affected by the type of data (parallel or text independent). Both methods perform better with parallel data than they do with text independent data but as the amount of data is increased the differences reduce for each of the two methods. The perceptual closeness between parallel and cross-lingual scenarios with large data is also reflected in listening tests which gave scores of only -0.17 (on a scale -2 to 2) in favor of the parallel case. Such a small difference between parallel and text independent results indicates a certain degree of efficiency of the proposed alignment scheme.

The contextual modeling is conceptually interesting and obtained slightly better results than the other methods. Our experiments answer the questions posed in section 6.3.2 showing that a contextual modeling can be better than models optimized globally on the full training data. We could not find clear evidence that the contextual modeling solved the over-smoothing problem. In fact, it could be argued that over-smoothing is partially caused by the MSE based criteria optimized by all these methods, in the sense that such criteria do not focus on details but on averages. In contextual modeling, however, this averaging is applied to a restricted 'local' dataset.

Future research could try finding and optimizing a perceptually motivated criterion or study new ways to separate style and content in speech, e.g. by modeling it as product of more than two underlying factors.

6.4 CONCLUSIONS

The objective in this chapter was to investigate spectral conversion techniques that could overcome current limitations leading to increased mapping accuracy and speech quality. This goal was pursued by proposing entirely new conversion approaches as well as improvement techniques for some existing conversion frameworks.

First of all, a new method based on bilinear models was proposed for the conversion of spectral envelopes. The method decomposes the LSF representation of the spectral envelope into a voice descriptive factor and a phonetic content factor and provides efficient training algorithms based on singular value decomposition (SVD). The method has a certain capability to reconstruct data unseen in the training set which was found to be similar to that of a conventional GMM-based conversion. The performance of the proposed bilinear approach has been found to be similar to the one offered by the GMM-based approach also otherwise in an extensive evaluation carried out over different types of training data. A side benefit of the proposed decomposition is that the speaker dependent factor could be useful in speaker recognition applications whereas the phonetic content factor could be a suitable parameterization in speech recognition where only the content is relevant.

The concepts of local or contextual models have also been introduced in this chapter. It is hypothesized that using local or contextual models trained from subsets of the training data can capture details better than a global fitting leading to improved accuracy and possibly alleviating the over-smoothing. A contextual modeling example was proposed in which the data subsets correspond to phonetic units situated before and after the frame to be converted. This modeling produced slightly smaller errors than the models trained globally. In addition to that, the local modeling was demonstrated using linear transformations trained locally. This approach was found to outperform a GMM-based conversion with globally fitted GMM in a subjective listening test and to effectively reduce the over-smoothing. The evaluations demonstrate the validity of the two concepts although more efficient implementations should be possible. On the downside, both method use heavier

computation for conversion as the conversion models depend on the input vector and have to be estimated at runtime.

In this chapter, it has also been argued that vector quantization can offer a flexible framework for voice conversion that has not been exploited to its full potential. A first improvement proposed to this framework is the memory efficient scheme based on multi-stage vector quantization described in section 6.1. Experimental results indicated both an improvement in accuracy and significant memory reduction compared to some prior art methods. Secondly, the local linear transformation technique is easily integrated with the same framework.

In the next subsections two directions for future research are proposed. The first direction is to integrate an existing hybrid technique based on GMM and FW with the proposed parametric speech model and to define an automatic procedure for formant alignment and for calculating the frequency warping function in a data driven manner. The second direction is to use dynamic programming and delta LSF information for optimizing the temporal evolution of the converted speech.

6.4.1 Future Work Proposal (1): Hybrid GMM-Frequency Warping

GMM and frequency warping techniques are commonly used in voice conversion. The frequency warping method introduced in section 2.4.3 is known to produce a high speech quality but fails to achieve a good identity mapping. On the other hand, the GMM method behaves well in terms of identity mapping but produces a clearly lower speech quality due to over-smoothing, formant broadening, etc. A combination of the two techniques in a hybrid approach is a natural way to eliminate the drawbacks of each method and to ensure both high speech quality and a good identity mapping. Such an approach has been proposed in [58][136]. Unfortunately, the existing frequency warping methods[22][57], including the hybrid approaches [58][136], were developed mainly on uncompressed speech. In frequency warping the goal is to find a warping function such that the spectral distance between the frequency-warped envelope of the source speaker and the envelope of the target speaker is minimized. In [57] [58], the warping function appears to be derived using heuristic and manual selection on the formants of the aligned spectral envelopes. This may hinder other applications where on demand specifications are necessary. Automatic procedures to derive the warping function have been proposed in [22] and [136] for discrete and continuous representations of the spectral envelope, respectively. The automatic mapping of formants introduced in [136] for an all-pole representation, uses exhaustive search to determine a set of pole pairs defining an optimal piece-wise linear warping function. The computation cost is increased by the fact that the number of frequency pairs is unknown and has to be determined as a part of the search procedure. The heavy computation represents a drawback of this method.

To summarize, the combined frequency warping and GMM methods reported in the literature [58][136] have the following drawbacks: 1) they cannot be directly applied in coded speech or would be less efficient; 2) in [58] the details of the warping function derivation are not reported, implying the possible manual selection of formant alignment; on the other hand, the automatic mapping of formants proposed in [136] relies on a computationally intensive exhaustive search procedure. 3) As a pitch-asynchronous scheme is used, special phase manipulation procedures are needed [58]; According to the authors' best knowledge, more efficient hybrid voice conversion approach that could avoid these problems have not been published anywhere.

The technique proposed in this section introduces an entire framework on hybrid approach on coded speech. The novel ideas are mainly in applying frequency warping into coded speech since GMM approach has already been reported. Another important aspect of the technique is the proposed automatic derivation of warping functions from aligned source and target spectra. Lending itself to efficient compression the proposed solution is very flexible and efficient and can be used in a variety of communications related and embedded applications. For instance, the technique can be used to create a high quality personalized voice conversion from a HQ-TTS voice since such data is typically stored in parameterized format due to memory limitations.

In the proposed hybrid approach, the GMM model is trained on a set of aligned joint LSF vectors of source and target speakers. For each GMM mixture, the mean vector is split into source and target parts and used to generate source and target spectral envelopes. Constrained search based on dynamic programming is proposed to automatically find the formant alignment for the pair of spectral envelopes. Then the warping function of each mixture is derived by curve fitting through the aligned formants. The particular warping function applicable to any given source frame in the conversion process, is a weighted combination of all mixture-specific warping functions. Posterior probabilities are used as the weights in the combination. Finally, the warping function is applied on the spectral envelope directly. More details can be seen in the following subsection.

Informal listening tests have shown that the converted speech performs well in terms of speaker identity, while the speech quality is kept at a high level.

Hybrid conversion for parametric speech

In our implementation we use the speech representation described in section 3.1.1 consisting of five features extracted at equal intervals from the speech signal:

1. LSFs (*lsf*), vocal tract contribution modeled using linear prediction;
2. Energy (*e*), overall gain;
3. Amplitude (*a*) of the sinusoids of excitation spectrum;
4. Pitch (*p*);
5. Voicing information (*v*);

A first component of the hybrid system introduced here is represented by a GMM trained on aligned data from the source and target speakers following the joint density approach presented in subsection 2.4.2.

For the l^{th} mixture component mean, (μ_l^x, μ_l^y), source and target spectral envelopes can be derived from the mean LSF vectors μ_l^x and μ_l^y. The aligned formants (more precisely, peaks) of the paired spectral envelopes are used to establish the mixture frequency warping function $W_l(\omega)$. Given a source feature vector x, the mixture weight, $p_l(x)$, can be calculated as in equation (2.21) from section 2.4.2. Then warping function for a given source feature vector x is obtained by a weighted combination of all mixture warping functions.

$$W(\omega) = \sum_{l=1}^{L} p_l(x) \cdot W_l(\omega) \tag{6.29}$$

Since the speech is coded as parameters, the warping function is applied to the LSF feature vector as shown in Figure 6.15. LSF feature vector of source speaker is firstly converted into linear prediction coefficient (LPC) vector. The spectral envelope is then obtained from LPC vector. The warping function is applied directly on the spectral envelope leading to the warped spectrum

$S(W^{-1}(\omega))$. The warped LPC vector is approximated from the warped spectrum, and the warped LSF feature vector is finally obtained from the warped LPC vector. The warping residual is introduced when estimating warped LPC from warped spectrum. The warped LSF, LSF_w, is output as converted LSF feature vector. The warping residual and warped excitation, together, form the generalized excitation.

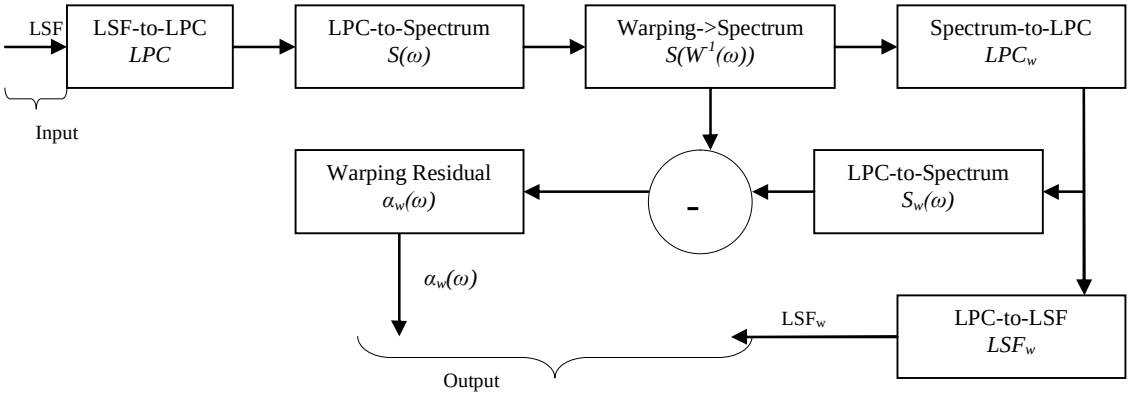

Figure 6.15: Algorithmic illustration of frequency warping on LSF feature vector.

Broadly speaking from speech production perspective, the speech S is generally modeled as vocal tract transfer function H (by LSF parameters) and excitation E (by amplitude parameters) as shown in Figure 6.16.

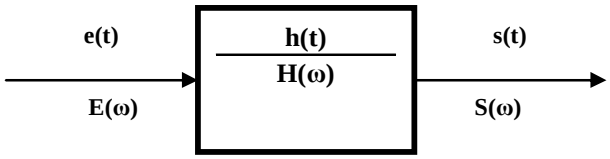

Figure 6.16: Speech production model.

As seen in equation (6.30), the speech is modeled in warped domain. The warped speech spectrum is the product of warped LPC spectrum and generalized excitation spectrum. The generalized excitation as shown in equation (6.31) is composed of warped excitation, warping residual and warped LPC spectrum. Weight, $1 \geq \lambda \geq 0$, is used to balance the contribution of the warping residual to the generalized excitation.

$$\begin{aligned} S(W^{-1}(\omega)) &= H(W^{-1}(\omega)) \cdot E(W^{-1}(\omega)) \\ &= \left[H_{lpc_w}(\omega) + \alpha_w(\omega)\right] \cdot E_w(\omega) \\ &= H_{lpc_w}(\omega) \cdot \left[1 + \frac{\alpha_w(\omega)}{H_{lpc_w}(\omega)}\right] \cdot E_w(\omega) \\ &= H_{lpc_w}(\omega) \cdot \tilde{E}_w(\omega) \end{aligned} \tag{6.30}$$

$$\tilde{E}_w(\omega) = \left[1 + \lambda \cdot \frac{\alpha_w(\omega)}{H_{lpc_w}(\omega)}\right] \cdot E_w(\omega) \tag{6.31}$$

where $H_{lpc_w}(\omega)$ is the transfer function of the LPC vector approximated from the warped spectrum; $\alpha_w(\omega)$ is the warping residual obtained as a difference between the warped spectrum and $H_{lpc_w}(\omega)$ (see Figure 6.15); $E_w(\omega)$ and $\tilde{E}_w(\omega)$ represent the warped excitation and warped generalized excitation, respectively.

We also propose a new method to automatically derive the frequency warping function. Given aligned LPC vectors from both source and target speakers, the corresponding spectral envelopes can be obtained. Suppose we have spectral peaks from source spectral envelop denoted as SP_1, SP_2, ..., SP_m, and from target spectral envelop denoted as TP_1, TP_2, ..., TP_n. A grid or lattice is generated, and each node denotes one possible alignment pair as shown in Figure 6.17. The aligned formant pairs are calculated using constrained search. The cost is defined for each node, and the path cost is the cumulative node cost for all the nodes in the path. The best path is the one with minimum path cost, as seen in equation (6.32).

$$path^* = \underset{path}{\text{argmin}} \sum_{i \in path} cost(i) \tag{6.32}$$

Then the warping function calculation is becoming to find the best path in the lattice. The node cost can be defined in different ways, for example of formant likelihood using peak parameters (shaping factor, peak bandwidth). In our current non-optimized implementation, we simply define the node cost as the distance to a baseline function (red line in Figure 6.17). It assumes that warping function has normally a minimal bias from the baseline function due to physiological limitations. By finding the best path, the formant pairs are determined. Thus the warping function is easily obtained by fitting a smooth curve through the aligned pairs.

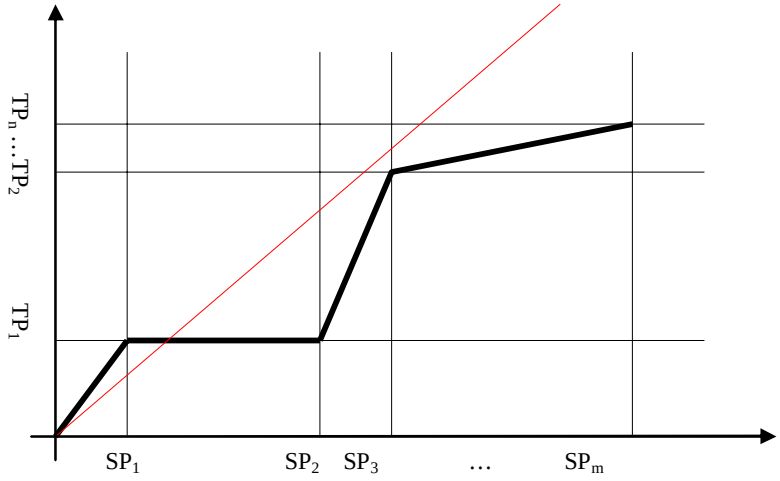

Figure 6.17: Formant alignment lattice.

For getting reliable mixture warping functions, the GMM model can be trained exclusively on the voiced data selected according to the voicing parameter in the feature set. The voicing value here is for the whole spectrum, computed using some additional logic based on the voicing values for the individual sinusoids that we used in equation (3.2) from subsection 3.1.1.

The Gaussian model is trained on the LSF data from speech segments that are fully voiced simultaneously in the source and in the target data.

Advantages and disadvantages

The proposed technique has a number of advantages. Similarly to [58], the method can achieve a good identity conversion and very good speech quality compared to the GMM-based approach. In addition, the method is fully data driven and can be applied directly on the coded speech in the parametric domain. The proposed technique is flexible and compatible with other existing speech coding applications. It is possible, for example, to use the technique in speech synthesis to modify a TTS output in order to obtain a customized voice. The procedure for automatic derivation of the warping function based on dynamic programming is likely to offer a significant computational advantage in comparison with the exhaustive search procedure proposed in [136]. Especially when used with the VLBR codec described in subsection 3.1.1 the method has the possibility to achieve a low overall computational complexity avoiding the conversion form an HSM representation to LSF and the special phase manipulation required in [58] and [136]. Due to the compact representation and the use of GMMs the method has also the potential to achieve a low memory footprint being an ideal solution for embedded applications.

On the downside, in contrast to [136] the proposed technique does not provide any mechanism for a correct conversion of the formant amplitudes or, in other words, does not control the energy distribution of the frequency warped spectrum.

6.4.2 Future Work Proposal (2): Dynamic Programming Optimization of Temporal Continuity

Most of the existing techniques do not model the temporal structure of speech although it can be argued that this is an important reason for quality degradation. Codebook mapping is one of the oldest techniques used for spectral conversion. In spite of its attractive properties in terms of detail preservation, its success has been very limited due to various shortcomings discussed in subsection 2.4.1. Many ideas proposed in the literature to improve its spectral continuity are shown in 2.4.1 to still suffer from different shortcomings. Except for [121] and [122], none of these techniques models the temporal information explicitly. The positive results reported in [121] are encouraging our expectation that further improvements can be achieved by finding alternative ways to model the speech dynamics.

This section presents some findings and proposes ideas for further investigations aiming to improve the temporal speech structure in vector quantization based voice conversion. First, a dynamic programming scheme is proposed to guide the codeword selection using delta features. This technique bears some similarity with [121]. Secondly, we propose a method to find an optimal sequence of static features based on dynamic information in the least square sense. This could be used as an extension to the dynamic programming scheme.

A research note

An interesting experimental result confirms our hypothesis that the nearest neighbor mapping is inaccurate and that the correct mapping should be sought among the K nearest neighbors. Given an input sequence $[x_1, \dots, x_T]$ we find for each x_t its K closest source code-words indexed $cx_1^t, \dots, cx_K^t$ by their increasing distance to x_t and list their target pairs $cy_1^t, \dots, cy_K^t$. If we convert the input

sequence by basic codebook mapping (or nearest neighbor approach) we would obtain $[cy_1^1, \ldots, cy_1^t, \ldots, cy_1^T]$.

Let $[y_1, \ldots, y_T]$ be the parametric sequence of a parallel target utterance which was time-aligned to the input speech. Let us consider its quantized representation as a sequence of target code-words $[cy^1, \ldots, cy^t, \ldots, cy^T]$. Our experiment indicates only a 20 percent probability that cy^t equals cy_1^t but for a relatively small K the probability that $cy^t \in [cy_1^t, \ldots, cy_K^t]$ grows to over 90 percent. This result clearly indicates not only that the basic nearest neighbor mapping is wrong but also suggests that a reasonably natural sound can be obtained by picking the right codeword out of K nearest neighbors. Next we will show how delta information can be used to help this selection process and we propose a dynamic programming scheme to achieve this goal.

Optimized temporal evolution using dynamic programming

To reflect the use of delta information our code-words will have the form $[{cx_k}^T, {\delta cx_k}^T]^T$ and $[{cy_k}^T, {\delta cy_k}^T]^T$ for source and target speaker respectively. The averaging effects associated with quantization are particularly undesirable for deltas therefore we will use as code-words real data points and real deltas.

Like in the previous research note, given the input sequence $[x_1, \ldots, x_T]$ we want to determine the converted result in the form of a target code-word sequence $[cy^1, \ldots, cy^t, \ldots, cy^T]$ where cy^t has to be selected from $[cy_1^t, \ldots, cy_K^t]$. There is a total of K^T (K to power T) possible output sequences but the question is which one explains best the associated delta information $[\delta cy^1, \ldots, \delta cy^t, \ldots, \delta cy^T]$. This can be formulated as an optimization problem and solved with dynamic programming.

Given a speech sequence $[y_1, \ldots, y_T]$, one way to define δy_t is:

$$\delta y_t = y_{t+1} - y_t \tag{6.33}$$

With δy_t defined in this way, we aim to select the output sequence $[cy^1, \ldots, cy^t, \ldots, cy^T]$ that minimizes the following error criterion:

$$E = \sum_{t=2}^{T} \| cy^t - (cy^{t-1} + \delta cy^{t-1}) \| \tag{6.34}$$

In dynamic programming terms this corresponds to a problem with K states corresponding to K possibilities to pick a code-word at every time instant t. This can be represented in the form of a trellis structure.

In order to find the minimum error path we need to move along the trellis one step at a time along the time axis, e.g. from time t to time t+1, keeping track at each moment t of the K best paths ending at moment t with each of the K code-words $cy_1^t, \ldots, cy_K^t$. The corresponding minimum costs of these paths will be denoted as $\varphi_1^t, \ldots, \varphi_K^t$. The cost of adding the codeword $cy^{t+1} = cy_l^{t+1}$ to a selected path $[cy^1, \ldots, cy^t = cy_k^t]$ can be defined as:

$$\varsigma_{k,l}^{t+1} = \left\| cy_l^{t+1} - (cy_k^t + \delta cy_k^t) \right\| \tag{6.35}$$

The minimum cost φ_l^{t+1} of a path ending at moment t+1 in $cy^{t+1} = cy_l^{t+1}$ can be defined recursively:

$$\varphi_l^{t+1} = \min_k \left[\varphi_k^t + \varsigma_{k,l}^{t+1} \right] \tag{6.36}$$

The algorithm to find the optimal path can be summarized as follows.

Algorithm – Dynamic Program for Optimal Path Selection

1. Initialization

$$\varphi_l^1 = 0$$
$$\varepsilon_l^1 = l \qquad (6.37)$$
$$\text{for } l = 1,2,\dots,K$$

2. Recursion

$$\varphi_l^{t+1} = \min_{1\le k\le K}\left[\varphi_k^t + \varsigma_{k,l}^{t+1}\right]$$
$$\varepsilon_l^{t+1} = \operatorname{argmin}_{1\le k\le K}\left[\varphi_k^t + \varsigma_{k,l}^{t+1}\right] \qquad (6.38)$$
$$\text{for } \; l = 1,2,\dots,K \; \text{ and } \; t = 1,2,\dots,T-1$$

3. Termination

$$\varphi_l^T = \min_{1\le k\le K}\left[\varphi_k^{T-1} + \varsigma_{k,l}^T\right]$$
$$\varepsilon_l^T = \operatorname{argmin}_{1\le k\le K}\left[\varphi_k^{T-1} + \varsigma_{k,l}^T\right] \qquad (6.39)$$

4. Path Backtracking

$$optimal\,path = (l_1, l_2, \dots, l_T), \qquad (6.40)$$
$$\text{where } l_T = \operatorname{argmin}_{1\le l\le K} \varphi_l^T \text{ and}$$
$$l_t = \varepsilon_{t+1}(l_{t+1}), \;\; \text{for } \; t = T-1, T-2, \dots, 1$$

The formulation as a dynamic programming problem may change depending on the error criterion which in turn depends on how δ is defined.

Least squares solution

Next, we assume the existence of a converted result in the form of a target codeword sequence $cy = [cy^1, \dots, cy^t, \dots, cy^T]$ with the attached delta information $\delta cy = [\delta cy^1, \dots, \delta cy^t, \dots, \delta cy^T]$ and we define $CY = [cy^1, \delta cy^1, \dots, cy^t, \delta cy^t, \dots, cy^T, \delta cy^T]$. These output sequences can but need not be computed with the previous path optimization scheme.

If the delta coefficients δcy^t are approximated as:

$$\delta cy^t = 0.5 \cdot (cy^{t+1} - cy^{t-1}) \qquad (6.41)$$

and the matrix W is defined as in Figure 6.18, the next relationship should hold.

$$CY = W \cdot cy \qquad (6.42)$$

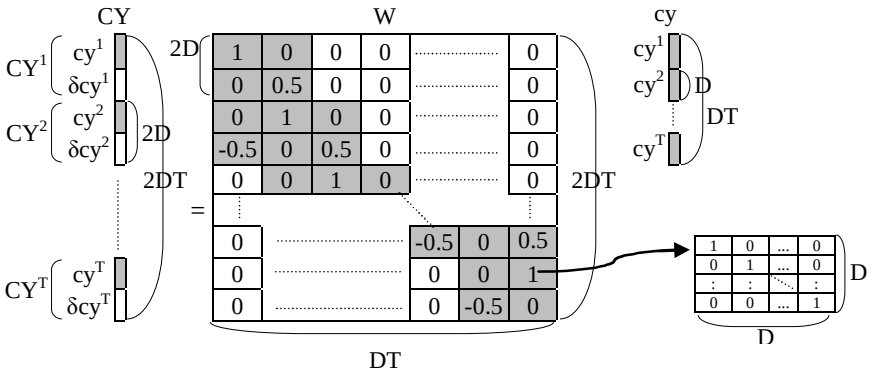

Figure 6.18: The relationship between a sequence of static feature vectors y and a sequence of static and dynamic feature vectors Y. (adapted from [100])

The delta coefficients represent important information which can be used to refine the converted result cy. This can be done by solving the equation (6.42) for the unknown cy in the least square sense with the solution:

$$\hat{y} = W^{+} \cdot CY \tag{6.43}$$

where $W^{+} = (W^{T}W)^{-1}W^{T}$ represents the pseudo-inverse of W and $\hat{y} = [\hat{y}_1^T, \hat{y}_2^T, \dots, \hat{y}_T^T]^T$ represents an improved estimation of the converted result.

This method is an effective way to refine the converted result of a codebook approach by making use of the temporal information (deltas). The idea can also be used iteratively by quantizing the new estimates $\hat{y} = [\hat{y}_1^T, \hat{y}_2^T, \dots, \hat{y}_T^T]^T$ and repeating the procedure with the new sequence of code-words (static features + their delta coefficients). It would be interesting to study the convergence of this process.

Chapter 7

Conclusions and Future Directions

The objective of this thesis was to develop a stand-alone voice conversion system based on a parametric speech representation and to study and propose algorithms in order to improve the quality and versatility of the existing solutions to stand-alone voice conversion. The work starts from the improvable aspects identified in the state of the art analysis carried out in Chapter 2. In spite of fairly successful results of current systems there is still room for improvement in all sub-areas of voice conversion towards providing both excellent quality and highly successful identity conversion. Most high-quality systems use a time-frequency speech representation. In the time domain, the frames are modeled and processed independently. Without a model of the temporal evolution of speech parameters these systems cannot ensure a smooth and natural sounding result. In the frequency domain, the inefficiency of spectral mapping can take various forms. Frequency warping has problems in controlling the shape of the modified formants while GMM-based conversion and other statistical methods are likely to suffer from over-smoothing. In terms of alignment there has been a growing interest for text-independent use cases due to practical reasons. These cases are more challenging and have typically achieved poorer results than the text-dependent scenarios. Yet another challenge addressed in this thesis is to find a parametric speech model that would allow to use voice conversion with coded speech and to integrate it with other communications related and embedded applications. The thesis covers three directions focusing on the spectral conversion as a fundamental task of voice conversion but treating in separate chapters also the parametric speech model and the alignment issue. Contributions have been made to all main parts of a voice conversion system and the conclusions will be presented next.

In Chapter 3 a parametric speech model inspired from speech coding is proposed as analysis-modification-synthesis framework for our voice conversion system. This representation offers efficient compression and compatibility with other speech coding solutions and embedded applications. The experiments demonstrate that the proposed model is suitable for the purpose of voice conversion allowing flexible speech manipulations. Furthermore, since the same parameterization is used internally by a TTS system, the voice conversion algorithms can be easily applied to customize the TTS output. In the conversion process the voicing level of a given spectral envelope may suffer undesired modifications becoming inconsistent with the voicing feature and

leading to quality degradations. The proposed voicing control scheme adjusts the voicing parameter to compensate for the voicing change caused by the conversion. Informal listening tests indicate that the scheme effectively reduced the noise level in the output samples improving the perceived quality. The speech enhancement technique presented in the final section of Chapter 3 offers an efficient way to manipulate the energy feature of the proposed speech model in order to attenuate the noise perceived during the pause portions of the converted signal. The method offers low complexity and the possibility for real-time implementation and its efficiency in terms of quality enhancement has been confirmed by informal listening tests. The data collection scheme introduced in subsection 3.4.1 shows how the parametric speech representation could be exploited in a phone application with the purpose of collecting speech data and continuously refining voice conversion models. The technique offers an enhanced user experience and the possibility to train high quality conversion models but its implementation details are omitted in this thesis leaving it as a potential future direction. Chapter 3 aimed to demonstrate the validity of the proposed speech representation for the voice conversion task and focused on the conversion of only pitch and line spectral frequencies. In order to achieve high-quality conversion results, future work should also consider the conversion of other features like residual signal, voicing or energy which contribute to a lesser degree to the speaker identity. Moreover, to do this correctly it is important to understand the interdependence between different features and preserve it during the conversion process.

Regarding the alignment, this thesis aimed to propose alternatives to existing techniques in order to improve the alignment accuracy and increase the system versatility. For the parallel case, a one-to-many frame alignment based on soft probabilities is shown to have several advantages over conventional hard alignment such as DTW. The validity of the idea is demonstrated with experimental results based on artificial data. Future work towards a full implementation should take into account the fact that for typical frame rates of analyzed speech and conventional 3-state phone HMMs the alignment probabilities approach binary decisions. To alleviate this problem and take full advantage of the idea, softer probabilities could be generated e.g. by using higher frame rates.

For their practical advantages, the text independent use cases have been attracting an increasing interest although their alignment also poses greater challenges. The first technique for text independent alignment proposed in Chapter 4 uses phonetic segmentation and time decomposition. Although the technique was not evaluated in comparison with an alternative text independent alignment it was successfully used in a real voice conversion application. The results obtained in listening tests for text-independent data aligned with this technique were found perceptually close to those obtained with a parallel data aligned with conventional DTW. A second approach to text-independent voice conversion is presented as a future direction without a full implementation or experimental results. The idea is to use a TTS system for reducing the problem to two concatenated conversions trained from parallel data.

The largest part of this thesis is dedicated to the conversion of spectral characteristics. The topic is treated in Chapters 5 and 6 with the aim of improving the mapping accuracy with respect to existing techniques without degrading the quality of the converted speech. Chapter 5 deals with a number of aspects related to the GMM-based voice conversion in order to upgrade the performance, efficiency and flexibility of this framework. The accuracy measure proposed in section 5.1 for the evaluation of GMM-based transformations has the advantage of fast computation and avoids the need for evaluation data. In the experiments, the technique was found in line with both perceptual observations and with MSE objective scores. The clustering and mode selection scheme introduced

in section 5.2 improves the conversion accuracy by grouping the training data into clusters with minimal variance of the target features and training a separate GMM for each class. This approach is data-driven and was shown in the experiments to outperform an equivalent conversion scheme that uses voicing-based clustering. In order to make this approach even more efficient, further consideration should be given to improving the class discrimination and to the compromise between the number of clusters and the reliability of the cluster model. In order to increase the system's flexibility and allow it to deal with small amounts of data, the technique proposed section 5.3 can be applied to adapt a well trained model to a new target speaker using limited data. In the experiments, a mean-adapted model performed clearly better than an equivalent EM model trained from scratch with the small data set. Subsection 5.4.1 presents an idea for improving the continuity and naturalness of the converted speech by including dynamic features in the estimated GMM model and optimizing an objective function defined by the converted static and dynamic features. Further work is needed towards the full implementation and verification of this idea which is given here only as a potential direction for future research.

In Chapter 6, spectral conversion is treated from a more general perspective and its improvement is sought outside the GMM framework. The proposed techniques are either completely new approaches or complementary to some existing frameworks. For example the method based on bilinear models has not been seen in the voice conversion literature while the idea of applying factor analysis in voice conversion is also very new and opening an entire line of future research. Factor analysis methods such as the bilinear models decompose the spectral envelope representation into a product of two or more underlying factors corresponding to either voice identity or phonetic content. The benefits of such decomposition extend beyond the area of voice conversion since the computed underlying factors could constitute suitable features for areas like speech or speaker recognition. In the evaluations, the proposed bilinear approach is found to have perceptually similar performance to the popular GMM based conversion. In contrast to the GMM based approach whose number of components has been optimized, the proposed approach did not appear to benefit from or need a parameter tuning.

The concepts of contextual and local modeling introduced in the same chapter aim to reduce the conversion error by replacing global fitting of models with multiple models trained on the fly on subsets of the full data. These concepts are in line with the previous idea of clustering and mode selection and with the observation that the mapping accuracy can be increased by reducing the data variance. In contrast to the technique described in section 5.2, where the fuzzy classification can be regarded as a drawback, the examples illustrating the proposed concepts have clear rules for classification. The contextual modeling was illustrated with a concrete case in which the phonetic context determines a subset of the training data which is used to train a conversion model. The errors were slightly smaller than those obtained with global models. Local modeling was demonstrated using locally trained linear transformations achieving better perceptual scores and lower over-smoothing relative to a globally trained GMM based conversion. The experiments prove the validity of these ideas which open the way for further refinements in this direction.

A merit of this thesis is to rediscover vector quantization and extend its applicability in spectral conversion beyond codebook mapping. Vector quantization can constitute a flexible foundation on top of which various techniques can be developed complementing it and benefiting of its attractive properties in terms of detail preservation. In addition to the local linear transformation technique which can be easily implemented within this framework, a memory-efficient conversion scheme

based on MSVQ is proposed. The experimental results indicated for this method not only a significant memory reduction but also a better accuracy compared to some conventional codebook conversions. Another idea that can be easily integrated with the vector quantization framework is presented as a future direction in subsection 6.4.2 and uses dynamic programming to optimize the temporal continuity of the parameter tracks.

The hybrid GMM-frequency warping technique for parametric speech presented in 6.4.1 preserves the performance advantages of the original method [58] in terms of speech quality while being likely to have a clear computational advantage compared to [58] and[136] due to the efficient speech representation and automatic derivation of the warping function.

A number of interesting directions for future research, some of which have been already mentioned in the previous analysis, can be considered as follows:

- **Factor analysis:** Factor analysis methods could provide a powerful tool for the separation of speech into a speaker related component and a phonetic content component. The new features would greatly benefit also other areas of speech processing such as speech recognition and speaker identification.
- **Vector quantization:** Vector quantization is a flexible platform offering the possibility for diverse extensions in addition to the ones proposed in this thesis. The framework is particularly suitable for alternative implementations of local modeling which was shown to improve the conversion accuracy. Other example worth of further consideration related to affine transformations and how they can be applied to take advantage of this framework.
- **Optimal linear transformation:** In the local linear transformation technique it was found beneficial to estimate a band diagonal linear transformation instead of a full matrix. Particularly for small data it is important to work with sparse matrices and avoid estimating unreliable matrix elements. It could be useful to stick to those elements which correspond to highly correlated LSF components or alternatively to apply regularization procedures for finding an optimal sparse matrix that best describes the input-output relationship.
- **Local modeling:** The fact that local modeling improves the conversion accuracy is supported with experimental results by several techniques proposed in this thesis including the clustering and mode selection scheme, the local linear transformation and the contextual modeling. In order to improve the performances in this direction the challenges are to find robust methods that can cope with reduced local data or, alternatively, efficient clustering and discrimination techniques.
- **Temporal modeling:** The absence of a model for the temporal evolution of speech parameters is believed to be an important source for quality degradation in most of the existing systems which, in general, perform the conversion frame by frame. Although some ideas for temporal modeling have been already proposed in the literature, further improvements are still possible and believed to have a great potential for increasing the quality and naturalness of the converted speech.
- **Enhanced parameterization:** The currently used speech features do not represent ideally the speaker dependencies and often cause problems with the synthetic speech quality. In order to achieve more suitable features it could be useful to create speech models able to mimic more closely the human speech production. Alternatively, as already suggested, deeper insight into the factor analysis methods could also provide an interesting track in this direction.

References

[1] G. Fant, Acoustic theory of speech production, The Hague, The Netherlands: Mouton & Co. N. V., 1960.

[2] M. Horne, Prosody: theory and experiment, Dordrecht, Netherlands: Kluwer Academic Publishers, 2000.

[3] D. Klatt and L. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *Journal of the Acoustical Society of America,* vol. 87, no. 2, pp. 820-857, 1990.

[4] C. Gobl and A. Chasaide, "The role of voicequality in communicating emotion, mood and attitude," *Speech Communication,* vol. 40, no. 1-2, p. 189–212, 2003.

[5] D. Childers and C. Lee, "Vocal Quality Factors: Analysis, Synthesis, and Perception," *Journal of the Acoustical Society of America,* vol. 90, no. 5, 1991.

[6] C. Hamon, E. Mouline and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modifications of speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1989.

[7] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication,* vol. 9, no. 5-6, pp. 453-467, 1990.

[8] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Communication,* vol. 16, no. 2, pp. 175-205, 1995.

[9] M. Dolson, "The phase vocoder: a tutorial," *Computer Music Journal,* vol. 10, no. 4, pp. 14-27, 1986.

[10] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal,* vol. 45, pp. 1493-1509, 1966.

[11] J. A. Moore, "The use of the phase vocoder in computer music applications," *Journal of the Audio Engineering Society,* vol. 24, no. 9, pp. 717-727, 1978.

[12] E. B. George, An analysis-by-synthesis approach to sinusoidal modeling applied to speech and music processing, PhD Thesis, Georgia Institute of Technology, 1991.

[13] E. B. George and M. J. T. Smith, "An analysis-by-synthesis approach to sinusoidal modeling applied to the analysis and synthesis of musical tones," *Journal of the Audio Engineering Society,* vol. 40, pp. 497-516, 1992.

[14] E. B. George and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," *IEEE Transactions on Speech and Audio Processing,* vol. 5, no. 5, pp. 389-406, 1997.

[15] E. Moulines and W. Verhelst, "Time-domain and frequency-domain techniques for prosodic modification of speech," in *Speech coding and synthesis*, Elsevier Science B.V., 1995, pp. 519-555.

[16] B. P. Nguyen, Studies on spectral modification in voice transformation, PhD Thesis, School of Information Science, Japan Advanced Institute of Science and Technology, 2009.

[17] A. Syrdal, Y. Stylianou, L. Garrison, A. Conkie and J. Schroeter, "TD-PSOLA versus HNM in diphone based speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, Washington, USA, 1998.

[18] D. Erro, Intra-lingual and cross-lingual voice conversion using harmonic plus stochastic models, Barcelona, Spain: PhD Thesis, Universitat Politechnica de Catalunya, 2008.

[19] Y. Stylianou, Harmonic plus noise models for speech, combined with statistical methods for speech and speaker modification, PhD Thesis, Ecole Nationale Superieure des Telecommunications, 1996.

[20] Y. S. Hsiao and D. G. Childers, "A new approach to formant estimation and modification based on pole interaction," in *Proceedings of the Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, USA, 1996.

[21] H. Mizuno, M. Abe and T. Hirokawa, "Waveform-based speech synthesis approach with a formant frequency modification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Minneapolis, Minnesota, USA, 1993.

[22] H. Valbret, E. Moulines and J. P. Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, no. 2-3, pp. 175-187, 1992.

[23] R. McAulay and T. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech model," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, San Diego, California, USA, 1984.

[24] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744-754, 1986.

[25] Y. Stylianou, O. Cappe and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131-142, 1998.

[26] M. Abe, S. Nakamura, K. Shikano and H. Kuwabara, "Voice conversion through vector quantization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, Washington, USA, 1988.

[27] Z. Shuang, R. Bakis and Y. Qin, "Voice conversion based on mapping formants," in *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, 2006.

[28] T. Masuko, K. Tokuda, T. Kobayashi and S. Imai, "Speech synthesis using HMMs with dynamic features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Atlanta, Georgia, USA, 1996.

[29] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black and K. Tokuda, "The HMM-based synthesis system (HTS) version 2.0," in *6-th ISCA Workshop on Speech Synthesis*, Bonn, Germany, 2007.

[30] D. Sündermann, Text-independent voice conversion, München, Germany: PhD Thesis, Universität der Bundeswehr München, 2007.

[31] A. Kain, High resolution voice transformation, PhD Thesis, OGI School of Science and Engineering, 2001.

[32] H. Höge, "Project proposal TC-STAR - make speech to speech translation real," in *Proceedings of the Language Resources Evaluation Conference (LREC)*, Las Palmas, Spain, 2002.

[33] J. Hossom, A. Kain, T. Mishra, J. v. Santen, M. Fried-Oken and J. Staehely, "Intelligibility and modifications to dysarthric speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, China, 2003.

[34] K. Nakamura, T. Toda, H. Saruwatari and K. Shikano, "Speaking aid system for total laryngectomees using voice conversion of body transmitted artificial speech," in *Proceedings of the Interspeech*, Pittsburgh, USA, 2006.

[35] E. Eide and M. Picheny, "Towards pooled-speaker concatenative text-to-speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Tolouse, France, 2006.

[36] J. Nurminen, V. Popa, J. Tian and I. Kiss, "A parametric approach for voice conversion," in *Proceedings of TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, 2006.

[37] J. Nurminen, J. Tian and V. Popa, "Voicing level control with application in voice conversion," in *Proceedings of Interspeech (Eurospeech)*, Antwerp, Belgium, 2007.

[38] V. Popa, J. Nurminen and J. Tian, "Apparatus, method and computer program product for advanced voice conversion," US Patent Application 20080082320, April 2008.

[39] J. Nurminen, V. Popa, E. Helander and J. Tian, "Voice conversion training and data collection". US Patent 7813924, October 2010.

[40] V. Popa, J. Nurminen and M. Gabbouj, "A study of bilinear models in voice conversion," *Journal of Signal and Information Processing*, vol. 2, no. 2, pp. 125-139, 2011.

[41] J. Tian, J. Nurminen and V. Popa, "Soft alignment based on a probability of time alignment". US Patent 7505950, March 2009.

[42] J. Tian, V. Popa and J. Nurminen, "Method, apparatus and computer program product for providing text independent voice conversion," US Patent Application 20090094031, April 2009.

[43] J. Tian, J. Nurminen and V. Popa, "Efficient Gaussian mixture model evaluation in voice conversion," in *Proceedings of Interspeech (ICSLP)*, Pittsburgh, PA, USA, 2006.

[44] J. Nurminen, J. Tian and V. Popa, "Novel method for data clustering and mode selection with application in voice conversion," in *Proceedings of Interspeech (ICSLP)*, Pittsburgh, PA, USA, 2006.

[45] J. Tian, V. Popa and J. Nurminen, "Efficient model re-estimation in voice conversion," in *Proceedings of EUSIPCO*, Lausanne, Switzerland, 2008.

[46] J. Nurminen, V. Popa and J. Tian, "Method, apparatus and computer program product for providing voice conversion using temporal dynamic features". US Patent 7848924, December 2010.

[47] V. Popa, J. Nurminen and M. Gabbouj, "A novel technique for voice conversion based on style and content decomposition with bilinear models," in *Proceedings of Interspeech*, Brighton, UK, 2009.

[48] V. Popa, H. Silen, J. Nurminen and M. Gabbouj, "Local linear transformation for voice conversion," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012.

[49] J. Nurminen, J. Tian and V. Popa, "Memory efficient method for high-quality codebook based voice conversion," US Patent Application 20080147385, June 2008.

[50] J. Nurminen, H. Silen, V. Popa, E. Helander and M. Gabbouj, "Voice conversion," in *Speech enhancement, modeling and recognition - algorithms and applications*, InTech, ISBN 978-953-51-0291-5, 2012.

[51] L. M. Arslan, "Speaker transformation algorithm using segmental codebooks (STASC)," *Speech Communication*, vol. 28, no. 3, pp. 211-226, 1999.

[52] D. Sündermann and H. Ney, "VTLN-based voice conversion," in *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology*, Darmstadt, Germany, 2003.

[53] H. Duxans, A. Bonafonte, A. Kain and J. v. Santen, "Including dynamic and phonetic information in voice conversion systems," in *Proceedings of the Interspeech (ICSLP)*, Jeju Island, Korea, 2004.

[54] D. Sündermann, H. Höge, A. Bonafonte and H. Duxans, "Residual prediction," in *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology*, Athens, Greece, 2005.

[55] O. Turk and L. M. Arslan, "Robust processing techniques for voice conversion," *Computer Speech and Language*, vol. 20, no. 4, pp. 441-467, 2006.

[56] H. Ye and S. Young, "Quality-enhanced voice morphing using maximum likelihood transformations," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1301-1312, 2006.

[57] Z. W. Shuang, R. Bakis, S. Shechtman, D. Chazan and Y. Qin, "Frequency warping based on mapping formant parameters," in *Proceedings of the Interspeech (ICSLP)*, Pittsburgh, PA, USA, 2006.

[58] D. Erro and A. Moreno, "Weighted frequency warping for voice conversion," in *Proceedings of Interspeech*, Antwerp, Belgium, 2007.

[59] R. McAulay and T. Quatieri, "Speech transformation based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 6, pp. 1449-1464, 1986.

[60] E. B. George and M. J. T. Smith, "A new speech coding model based on a least-squares sinusoidal representation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, USA, 1987.

[61] J. Wouters and M. Macon, "Spectral modification for concatenative speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[62] H. Ye and S. Young, "High quality voice morphing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, 2004.

[63] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 8, pp. 664-678, 1988.

[64] T. Dutoit and H. Leich, "Text-to-speech synthesis based on a MBE re-synthesis of segments database," *Speech Communication*, vol. 13, pp. 435-440, 1993.

[65] Y. Stylianou, "A pitch and maximum voiced frequency estimation technique adapted to harmonic models of speech," in *Proceedings of the IEEE Nordic Signal Processing Symposium*, Helsinki, Finland, 1996.

[66] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Munich , Germany , 1997.

[67] A. S. Spanias, "Speech coding: a tutorial review," *Proceedings of the IEEE*, vol. 82, no. 10, pp. 1541-1582, 1994.

[68] T. Drugman and T. Dutoit, "The deterministic plus stochastic model of the residual signal and its applications," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 968-981, 2012.

[69] J. Makhoul, "Linear prediction: a tutorial review," in *Proceedings of the IEEE*, 1975.

[70] J. Makhoul and J. Wolf, "Linear prediction and the spectral analysis of speech," BBN Report No. 2304, 1972.

[71] J. D. Markel and A. H. Gray, Linear prediction of speech, New York: Springer-Verlag, 1976.

[72] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *Journal of the Acoustical Society of America*, vol. 50, no. 2, pp. 637-655, 1971.

[73] F. Itakura and S. Saito, "Speech analysis/synthesis by partial correlation coefficient," in *Proceedings of the 4th Electrical Engineers Society Joint Conference*, 1970.

[74] F. Itakura and S. Saito, "On the optimum quantization of feature parameters in the PARCOR speech synthesizer," in *Proceedings of the IEEE Conference on Speech Communication and Processing*, 1972.

[75] F. K. Soong and B. H. Juang, "Line spectrum pair (LSP) and speech data compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, San Diego, California, USA, 1984.

[76] F. K. Soong and B. H. Juang, "Optimal quantization of LSP parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 15-24, 1993.

[77] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *IEEE Transactions on Acoustics, Speech and Signal Processing* , vol. 34, no. 6, pp. 1419-1426, 1986.

[78] B. Atal and M. R. Schroeder, "Linear prediction analysis of speech based on a pole-zero representation," *Journal of the Acoustical Society of America*, vol. 64, no. 5, p. 1310, 1978.

[79] N. Mikami and R. Ohba, "Pole-zero analysis of voiced speech using group delay characteristics," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 5, p. 1095, 1984.

[80] W. Mikhael, A. Spanias, G. Kang and L. Fransen, "Pole-zero prediction," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 3, p. 352, 1986.

[81] R. L. Miller, "Nature of the vocal cord wave," *Journal of the Acoustical Society of America*, vol. 31, no. 6, pp. 667-677, 1959.

[82] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153-165, 2010.

[83] J. L. Flanagan, Speech analysis, synthesis and perception, 2nd ed., New York: Springer-Verlag, 1972.

[84] P. Alku, "Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering," *Speech Communication*, vol. 11, no. 2-3, pp. 109-118, 1992.

[85] D. Wong, J. Markel and J. A. Gray, "Least squares glottal inverse filtering from the acoustic speech waveform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 4, pp. 350-355, 1979.

[86] H. Strube, "Determination of the instant of glottal closure from the speech wave," *Journal of the Acoustical Society of America*, vol. 56, no. 5, pp. 1625-1629, 1974.

[87] M. Plumpe, T. Quatieri and D. Reynolds, "Modeling of the glottal flow derivative waveform with application to speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 569-585, 1999.

[88] M. Fröhlich, D. Michaelis and H. Strube, "SIM—Simultaneous inverse filtering and matching of a glottal flow model for acoustic speech signals," *Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 479-488, 2001.

[89] O. Akande and P. Murphy, "Estimation of the vocal tract transfer function with application to glottal

wave analysis," *Speech Communication,* vol. 46, no. 1, pp. 15-36, 2005.

[90] Q. Fu and P. Murphy, "Robust glottal source estimation based on joint source-filter model optimization," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 14, no. 2, pp. 492-501, 2006.

[91] P. Alku, H. Tiitinen and R. Näätänen, "A method for generating natural-sounding speech stimuli for cognitive brain research," *Clinical Neurophysiology,* vol. 110, no. 8, pp. 1329-1333, 1999.

[92] H. Matsumoto, S. Hiki, T. Sone and T. Nimura, "Multidimensional representation of personal quality of vowels and its acoustical correlates," *IEEE Transactions on Audio and Electroacoustics,* vol. 21, no. 5, pp. 428-436, 1973.

[93] K. Itoh and S. Saito, "Effects of acoustical feature parameters of speech on perceptual identification of speaker," *IECE Transactions,* vol. J65, no. A, pp. 101-108, 1982.

[94] S. Furui, "Research on individuality features in speech waves and automatic speaker recognition techniques," *Speech Communication,* vol. 5, no. 2, pp. 183-197, 1986.

[95] H. Kuwabara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: control and conversion," *Speech Communication,* vol. 16, no. 2, pp. 165-173, 1995.

[96] H. Sato, "Acoustic cues of female voice quality," *Electronics and Communication in Japan,* vol. 57, no. A, pp. 29-38, 1974.

[97] K. Tokuda, T. Kobayashi, T. Masuko and S. Imai, " Mel-generalized cepstral analysis - a unified approach to speech spectral estimation," in *Proceedings of ICSLP*, Yokohama, Japan, 1994.

[98] S. Desai, A. Black, B. Yegnanarayana and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech and Language Processing* , vol. 18, no. 5, pp. 954-964, 2010.

[99] E. Helander, T. Virtanen, J. Nurminen and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 18, no. 5, pp. 912-921, 2010.

[100] T. Toda, A. Black and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 15, no. 8, pp. 2222-2235, 2007.

[101] K. Tokuda, H. Zen and A. Black, "An HMM-based speech synthesis system applied to English," in *Proceedings IEEE Workshop on Speech Synthesis*, Santa Monica, CA, USA, 2002.

[102] T. Toda, A. W. Black and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, Pennsylvania, USA, 2005.

[103] Y. Stylianou, "Voice transformation," in *Tutorial at Interspeech 2007*, Antwerp, Belgium, 2007.

[104] H. Duxans, D. Erro, J. Perez, F. Diego, A. Bonafonte and A. Moreno, "Voice conversion of non-aligned data using unit selection," in *TC-STAR Workshop on Speech to Speech Translation*, Barcelona, Spain, 2006.

[105] Y. Nankaku, K. Nakamura, T. Toda and K. Tokuda, "Spectral conversion based on statistical models including time-sequence matching," in *6th ISCA Workshop on Speech Synthesis*, Bonn, Germany, 2007.

[106] H. Ye and S. Young, "Voice conversion for unknown speakers," in *Proceedings of ICSLP*, Jeju Island, Korea, 2004.

[107] M. Abe, K. Shikano and H. Kuwabara, "Cross-language voice conversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Albuquerque, New Mexico, USA, 1990.

[108] M. Mashimo, T. Toda, K. Shikano and N. Campbell, "Evaluation of cross-language voice conversion based on GMM and STRAIGHT," in *Proceedings of Eurospeech*, Aalborg, Denmark, 2001.

[109] D. Sündermann, A. Bonafonte, H. Ney and H. Höge, "A first step towards text-independent voice conversion," in *Proceedings of ICSLP*, Jeju Island, Korea, 2004.

[110] D. Sündermann, H. Höge, A. Bonafonte, H. Ney, A. Black and S. Narayanan, "Text-independent voice conversion based on unit selection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006.

[111] J. Tao, M. Zhang, J. Nurminen, J. Tian and X. Wang, "Supervisory data alignment for text-independent

voice conversion," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 18, no. 5, pp. 932-943, 2010.

[112] D. Erro, A. Moreno and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 18, no. 5, pp. 944-953, 2010.

[113] L. M. Arslan and D. Talkin, "Speaker transformation using sentence HMM based alignments and detailed prosody modification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Seattle, Washington, USA, 1998.

[114] C. H. Lee and C. H. Wu, "MAP-based adaptation for speech conversion using adaptation data selection and non-parallel training," in *Proceedings of ICSLP*, Pittsburgh, PA, USA, 2006.

[115] A. Mouchtaris, J. Van der Spiegel and P. Mueller, "“Nonparallel training for voice conversion based on a parameter adaptation approach," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 14, no. 3, pp. 952-963, 2006.

[116] T. Masuko, K. Tokuda, T. Kobayashi and S. Imai, "Voice characteristics conversion for HMM-based speech synthesis system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Munich, Germany, 1997.

[117] M. Tamura, T. Masuko, K. Tokuda and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using MLLR," in *Proceedings of ESCA/COCOSDA Workshop on Speech Synthesis,* Jenolan Caves, Australia, 1998.

[118] Y. Kang, Z. Shuang, J. Tao, W. Zhang and B. Xu, "A hybrid GMM and codebook mapping method for spectral conversion," in *Proceedings of ACII,* Beijing, China, 2005.

[119] Z. W. Shuang, Z. X. Wang, Z. H. Ling and R. H. Wang, "A novel voice conversion system based on codebook mapping with phoneme-tied weighting," in *Proceedings of Interspeech*, Jeju Island, Korea, 2004.

[120] Y. P. Wang, Z. H. Ling and R. H. Wang, "Emotional speech synthesis based on improved codebook mapping voice conversion," in *Proceedings of ACII,* Beijing, China, 2005.

[121] M. Eslami, H. Sheikhzadeh and A. Sayadiyan, "Quality improvement for voice conversion systems based on trellis structured vector quantization," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[122] Ö. Salor and M. Demirekler, "Dynamic programming approach to voice transformation," *Speech Communication,* vol. 48, no. 10, pp. 1262-1272, 2006.

[123] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 39, no. 1, pp. 1-38, 1977.

[124] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Seattle, Washington, USA, 1998.

[125] S. Geman, E. Bienenstock and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Communication,* vol. 4, no. 1, pp. 1-58, 1992.

[126] L. Mesbashi, V. Barreaud and O. Boeffard, "Comparing GMM-based speech transformation systems," in *Proceedings of Interspeech*, Antwerp, Belgium, 2007.

[127] A. M. Kondoz, Digital speech coding for low bit rate communications systems, England: Wiley and Sons, 2004.

[128] T. Toda, H. Saruwatari and K. Shikano, "Voice conversion algorithm based on Gaussian mixture model with dynamic frequency warping of STRAIGHT spectrum," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Salt Lake City, Utah, USA, 2001.

[129] Y. Chen, M. Chu, E. Chang, J. Liu and R. Liu, "Voice conversion with smoothed GMM and MAP adaptation," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[130] H. Benisty and D. Malah, "Voice conversion using GMM with enhanced global variance," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[131] T. Dutoit, A. Holzapfel, M. Jottrand, A. Moinet, J. Perez and Y. Stylianou, "Towards a voice conversion system based on frame selection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Honolulu, Hawaii, USA, 2007.

[132] E. Helander, H. Silen, J. Miguez and M. Gabbouj, "Maximum a posteriori voice conversion using

sequential Monte Carlo methods," in *Proceedings of Interspeech*, Makuhari, Japan, 2010.

[133] B. Nguyen and M. Akagi, "Phoneme-based spectral voice conversion using temporal decomposition and gaussian mixture model," in *Proceedings of International Conference on Communications and Electronics*, HoiAn, Vietnam, 2008.

[134] D. Rentzos, S. Vaseghi, Q. Yan and C. H. Ho, "Voice conversion through transformation of spectral and intonation features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, 2004.

[135] D. Rentzos, S. Vaseghi, Q. Yan and C. H. Ho, "Parametric formant modelling and transformation in voice conversion," *International Journal of Speech Technology*, vol. 8, p. 227–245, 2005.

[136] D. Erro, A. Moreno and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 5, pp. 922-931, 2010.

[137] M. E. Lee, Acoustic models for the analysis and synthesis of the singing voice, PhD Thesis, Georgia Institute of Technology, 2005.

[138] M. Narendranath, H. A. Murthy, S. Rajendran and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, no. 2, pp. 207-216, 1995.

[139] G. Baudoin and Y. Stylianou, "On the transformation of the speech spectrum for voice conversion," in *Proceedings of ICSLP*, Philadelphia, PA, USA, 1996.

[140] T. Watanabe, T. Murakami, M. Namba, T. Hoya and Y. Ishida, "Transformation of spectral envelope for voice conversion based on radial basis function networks," in *Proceedings of ICSLP*, Denver, Colorado, USA, 2002.

[141] E. Helander, H. Silen, T. Virtanen and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 806 - 817 , 2011.

[142] P. Song, Y. Bao, L. Zhao and C. Zou, "Voice conversion using support vector regression," *Electronic Letters*, vol. 47, no. 18, pp. 1045-1046, 2011.

[143] H. Mori and H. Kasuya, "Speaker conversion in ARX-based source-formant type speech synthesis," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[144] D. Sündermann, H. Höge, A. Bonafonte, H. Ney and J. Hirschberg, "Text-independent cross-language voice conversion," in *Proceedings of ICSLP*, Pittsburgh, PA, USA, 2006.

[145] H. Duxans and A. Bonafonte, "Residual conversion versus prediction on voice morphing systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006.

[146] Z. Hanzlicek and J. Matousek, "F0 transformation within the voice conversion framework," in *Proceedings of Interspeech*, Antwerp, Belgium, 2007.

[147] W. S. Percybrooks and E. Moore II, "New algorithm for LPC residual estimation from LSF vectors for a voice conversion system," in *Proceedings of Interspeech*, Antwerp, Belgium, 2007.

[148] A. Rämö, J. Nurminen, S. Himanen and A. Heikkinen, "Segmental speech coding model for storage applications," in *Proceedings of ICSLP*, Jeju Island, South Korea, 2004.

[149] R. J. McAulay and T. F. Quatieri, "Sinusoidal coding," in *Speech coding and synthesis*, Elsevier Science B. V., 1995, pp. 121-174.

[150] R. J. McAulay and T. F. Quatieri, "Pitch estimation and voicing detection based on a sinusoidal speech model," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Albuquerque, New Mexico, USA, 1990.

[151] "Technology and Corpora for Speech to Speech Translation," [Online]. Available: http://www.tcstar.org.

[152] T. E. Tremain, "The government standard linear predictive coding," *Speech Technology Magazine*, pp. 44-49, 1982.

[153] W. B. Kleijn and J. Haagen, "Waveform interpolation for coding and synthesis," in *Speech Coding and Synthesis*, Elsevier Science B. V., 1995, pp. 175-207.

[154] E. Helander and J. Nurminen, "A novel method for prosody prediction in voice conversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, 2007.

[155] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113-120, 1979.

[156] B. S. Atal, "Efficient coding of LPC parameters by temporal decomposition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Boston, Massachusetts, USA, 1983.

[157] C. N. Athaudage, A. B. Brabley ja M. Lech, ”Optimization of a temporal decomposition model of speech,” tekijä: *Proceedings of the International Symposium on Signal Processing and Its Applications (ISSPA)*, Brisbane, Australia, 1999.

[158] M. Niranjan and F. Fallside, "Temporal decomposition: a framework for enhanced speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, Arizona, USA, 1989.

[159] P. J. Dix and G. Bloothooft, "A breakpoint analysis procedure based on temporal decomposition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 9-17, 1994.

[160] P. C. Nguyen, T. Ochi and M. Akagi, "Modified restricted temporal decomposition and its application to low bit rate speech coding," *IEICE Transactions on Information Systems*, Vols. E86-D, p. 397–405, 2003.

[161] P. C. Nguyen, M. Akagi and T. B. Ho, "Temporal decomposition: a promising approach to VQ-based speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, Hong Kong, 2003.

[162] B. P. Nguyen, T. Shibata and M. Akagi, "High-quality analysis/synthesis method based on temporal decomposition for speech modification," in *Proceedings of Interspeech*, Brisbane, Australia, 2008.

[163] T. Shibata and M. Akagi, "A study on voice conversion method for synthesizing stimuli to perform gender perception experiments of speech," in *Proceedings of the RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP)*, Gold Coast, Australia, 2008.

[164] J. Tian, J. Nurminen and V. Popa, "Method, apparatus, mobile terminal and computer program product for providing efficient evaluation of feature transformation". US Patent 7480641, January 2009.

[165] D. Sündermann, A. Bonafonte, H. Ney and H. Höge, "Voice conversion using exclusively unaligned training data," in *Proceedings of ACL/EMNLP*, Barcelona, Spain, 2004.

[166] A. Kumar and A. Verma, "Using phone and diphone based acoustic models for voice conversion: a step towards creating voice fonts," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, Hong Kong, 2003.

[167] J. Tian, J. Nurminen and V. Popa, "Method, apparatus, mobile terminal and computer program product for providing data clustering and mode selection". US Patent 7725411, May 2010.

[168] T. Toda, Y. Ohtani and K. Shikano, "Eigenvoice conversion based on Gaussian mixture model," in *Proceedings of ICSLP*, Pittsburgh, PA, USA, 2006.

[169] L. M. Arslan and D. Talkin, "Voice conversion by codebook mapping of line spectral frequencies and excitation spectrum," in *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[170] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmoud and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4kb/s speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 373-385, 1993.

[171] B. Bhattacharya, W. P. LeBlanc, S. Mahmoud and V. Cuperman, "Tree searched multi-stage vector quantization of LPC parameters for 4 kb/s speech coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, San Francisco, California, USA, 1992.

[172] H. Ye and S. Young, "Perceptually weighted linear transformations for voice conversion," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[173] W. T. Freeman, J. B. Tenenbaum and E. Pasztor, "Learning style translation for the lines of a drawing," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 33-46, 2003.

[174] "CMU ARCTIC speech synthesis databases," [Online]. Available: http://festvox.org/cmu_arctic/.

[175] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural computation*, vol. 12, no. 6, pp. 1247-1283, 2000.

[176] E. Helander, J. Nurminen and M. Gabbouj, "Analysis of LSF frame selection in voice conversion," in *Proceedings of the International Conference on Speech and Computer*, Moscow, Russia, 2007.

[177] E. Helander, J. Nurminen and M. Gabbouj, "LSF mapping for voice conversion with very small training sets," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, USA, 2008.

[178] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations," in *Proceedings of Eurospeech*, Madrid, Spain, 1995.

[179] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3-14, 1993.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-2835-4
ISSN 1459-2045