TR-I-0191

A STUDY ON SPEAKER INDIVIDUALITY CONTROL

音声の個人性制御の研究

Masanobu ABE
阿部匡伸

ATR Interpreting Telephony Labs.
(ATR自動翻訳電話研究所)

1991.1

Abstract

In this report, we will discuss algorithms to change speaker individuality: i.e., speech uttered by a speaker is changed or modified to sound as if another speaker had uttered it. First, we formulate voice conversion as a mapping problem by introducing vector quantization. Secondly, we propose a new algorithm which makes it possible to synthesize high quality speech even if the pitch frequency or duration is somewhat changed. Third, we discuss if speaker individuality can be controlled across different languages. Finally, to improve voice conversion performance, we propose also to convert the dynamic characteristics of speaker individuality by using speech segments as conversion units.

内容梗概

音声の個人性を変換する目的で行った研究である。まず、コードブックマッピングによる声質変換アルゴリズムを提案し、これを評価した。次に、短時間フーリエスペクトル分析合成系を用いたピッチ周波数変換アルゴリズムを提案し、これを評価した。第三に、バイリンガル話者の英語と日本語を分析し、言語による音声スペクトルの差を明らかにした。この分析結果に基ずいて、コードブックマッピングによる声質変換アルゴリズムを用いて、言語にわたる声質変換を試みた。最後に、コードブックマッピングによる声質変換アルゴリズムをフレーム単位からセグメント単位に拡張し、声質変換の高度化を試みた。

Contents

CONTENTS

List of Figures

LIST OF FIGURES

LIST OF FIGURES

List of Tables

Acknowledgements

First and foremost, I would like to express my sincere appreciation to Dr. Kiyohiro Shikano and Dr. Hisao Kuwabara who recommended the thesis presentation, for their kind encouragement, guidance and discussions throughout this study.

I wish to express my sincere thank to Dr. Akira Kurematsu, President of ATR Interpreting Telephony Research Labs., for his kind encouragement and for giving me a chance to achieve this study.

I also owe a great deal to the members of the ATR Speech Processing Department. Satoshi Nakamura has closely worked with me on voice conversion based on codebook mapping. I would like thank him for his discussions and kind cooperation. I would like thank Shin'ichi Tamura for his discussions on signal processing. I would like thank Kenj Kita for providing the HMM-LR recognition system in segment-based approach. I also would like thank Shigeki Sagayama for his kind suggestions and discussions on segment-based approach. I would like to thank Drs. Yoshinori Sagisaka, Takeshi Kawabata, and Masahide Sugiyama for their kind encouragement and discussions.

I wish to express my sincere thank to Dr. Victor Zue, head of the LCS-SLS group at MIT, for his kind suggestions and discussions on cross-language voice conversion, and for giving me a chance to work with his group.

Chapter 1

Prologue

1. Introduction

Speech is the primary communication method for human beings. Only speech makes it possible to communicate without particular tools. In that sense, it is often said that speech is the most convenient and natural medium. Although this might be the primary benefit of speech, we unconsciously use other benefits of speech in daily speech communication. One of these is speech quality. As every person has a different face, speech sound uttered by a speaker has a particular quality specific to the speaker, which we call "speaker individuality".

This thesis reports a study of speaker individuality control. The goal of this research is to change speaker individuality: i.e., speech uttered by a speaker is changed so as to sound as if another speaker had uttered it. In the remaining section of this chapter, speaker individuality is discussed in detail, and the scope of this thesis is explained.

2. Speaker Individuality

2.1 The Role of Speaker Individuality

Speaker individuality plays an important role in smooth communication, and enriches our daily communication. When we converse over the telephone, for example, we always try to identify the speaker. If the speaker is a someone close such as a family member or a friend, we confirm him/her through speech quality. If this is different from what we expect, we assume he/she has a cold or is a different person.

When we make a vocal request, speaker individuality is important information. Imaging that a father and daughter are in different rooms, and that the father asks her to, "please bring my bag." The daughter would never bring her mother's bag, because she judges the speaker to be her father through speech quality. If speech did not contain any information on a speaker, we would always have to confirm the speaker by extra questions. This would be very bothersome.

Why do we enjoy a radio mystery? Because we can identify the characters by their speech quality, and that picture the story in our mind's eye. This is also true of radio programs such as panel discussions and interviews.

As shown in the above examples, speaker individuality is useful not only in identifying a speaker, but also helps us communicate smoothly.

2.2 Where Does Speaker Individuality Come From?

Speaker individuality is molded by both social and physiological factors. The social factors include the social environment in which a speaker grew up. An extreme example is a language. Even if all human beings have the same speech organ system, their acquired spoken-language attributes such as phonemes, accent, intonation and so on, differ considerably according to his/her environment. A dialect is another example. Social class also affects speaking style or a speaker's tendency to use certain words and syntactic structures.

The physiological factors are variability in speech organs among speakers. It is well known, for example, that the primary difference between children's speech and adult's speech arises from the size of their speech organs. It is easy to understand that the speech of blood relatives is very close in quality because of the physical similarity in their speech organs.

A speech production model was usually introduced to investigate speaker individuality. According to speech production theory, speech sound is generated by the periodic vibration of the vocal cord. The the peculiarities of vocal cord vibration and its periodicity are called "source characteristics". The air-flow passing through the vocal cord is modulated according to the shape

(configuration) of the vocal tract, which is modeled as a kind of resonant tube. The vocal tract has particular "resonant characteristics" for any given shape, and sounds such as different vowels and consonants, are produced as it changes shape.

Based on the speech production model, the acoustic cues of speaker individuality have been the subject of many studies. In terms of source characteristics, for example, large differences between male and female speech were reported[Price, 1989][Childers, 1985]. The differences in resonant characteristics such as formant frequencies, bandwidth, and spectrum tilt among speakers, have been reported[Sato, 1974][Itoh, 1982][Furui, 1985][Kuwabara, 1987] [Klatt, 1990] and utilized in speaker recognition[Rosenberg, 1976][Furui, 1981][Soong, 1988]. However, not all acoustic cues of speaker individuality have been explained. It is now commonly accepted that speaker individuality resides not in a single feature, but is distributed over various acoustic features, and features which characterize speaker individuality are difficult to separate from features which characterize phonemes.

3 Thesis Scope

In this thesis, we will discuss algorithms to change speaker individuality: i.e., speech uttered by a speaker is changed or modified to sound as if another speaker had uttered it, which we call "voice conversion". What we try to change is only the speaker individuality which arises from physiological factors.

From point of view of a speaker individuality control, not only must we extract parameters which characterize a speaker, but we must also formulate parameter conversion rules between speakers. Even if both problems were essentially solved, speech quality would not always be changed successfully because of the correlation between parameters. Both formant frequency shift and formant bandwidth modification, for example, will cause changes in spectrum tilt and vice versa[Takagi, 1987][Hakoda, 1987]. Parameter changes in glottal volume velocity function cause a shift in the first formant frequency and a change in its band width. Generally speaking, it is very difficult to reasonably control

parameters including these side effects. In chapter 2, to avoid such difficulties, we formulate voice conversion as a mapping problem by introducing vector quantization. An advantage of this approach is that features which represent speaker individuality are not extracted explicitly, but implicitly.

Because pitch frequency is important information on speaker individuality, it is necessary to change pitch frequency for speaker individuality control. Although the modification of pitch frequency is possible using conventional vocoder algorithms, the modified speech quality is not good enough. In chapter 3, we propose a new algorithm which makes it possible to synthesize high quality speech even if the pitch frequency or duration is somewhat changed.

To take social factors into account, spoken-language is considered to be an aspect of speaker individuality. If speaker individuality can be controlled across different languages, in other words, if English can be synthesized as if a Japanese speaker uttered it, it would be very useful. In chapter 4, we discuss the possibility and perform some experiments.

The proposed algorithm in chapter 2 makes it possible to convert only the static characteristics of speaker individuality. In chapter 5, to improve voice conversion performance, we propose also to convert the dynamic characteristics of speaker individuality by using speech segments as conversion units. The importance of the dynamic characteristics of speaker individuality is discussed.

Chapter 2

Voice Conversion Based on Codebook Mapping

1. Introduction

Speech individuality generally consists of two major factors: acoustic features and prosodic features. In this chapter, we are going to discuss control of the acoustic features. To control speech individuality, we have to know which parameters are most important for representing the speaker, and how to control these parameters. It is, however, difficult to ansewer such questions by analyzing speech data, because speech individuality is distributed among various parameters such as formant frequencies and bandwidths, spectral tilt, and glottal waveforms [Rosenberg, 1976] [Furui, 1981] [Childers, 1985] [Kuwabara, 1987] [Price, 1989]. To solve the voice conversion problem in a sophisticated manner, we formulate it as a mapping problem by introducing vector quantization [Shikano, 1986] [Abe, 1988].

Figure 2.1 shows the basic idea of the voice conversion using vector quantization. Ellipses in Fig. 2.1 represent codebooks (spectrum spaces) of speaker A and speaker B, and the black dots in each ellipse are code vectors (speech spectra). Let's try to convert speaker A's speech to speaker B's speech. If the code vectors in these codebooks have one-to-one correspondences, like code vector A1 and code vector B1 in Fig. 2.1, voice conversion is easily performed by replacing A1 with B1. In other words, a conversion of acoustic features from one speaker to another is reduced to the problem of finding a correspondence between the codebooks of the two speakers. However, as with A2-A3 and B2-B5, code vectors usually dose not have one-to-one correspondences. Therefore, we would like to generate a new codebook whose code vectors have a one-to-one correspondence with speaker A's code vectors. We call this new codebook a "mapping codebook".

In section 2, vector quantization technique is briefly reviewed. In section 3, a method of making mapping codebooks and a synthesis procedure are described. In section 4, the performance of the proposed algorithms are evaluated by measuring distortion and listening tests. In section 5, improved algorithms are proposed and evaluated.

2. Vector Quantization[Makhoul, 1985]

The conversion of an analog source into a digital source consists of two parts: sampling and quantization. Sampling converts a continuous-time signal into a discrete-time signal by measuring the signal value at regular time intervals. Quantization converts a continuous-amplitude signal into a set of discrete amplitudes. When each of a set of parameters is quantized separately, the process is known as scalar quantization. When the set of parameters is quantized jointly as a single vector, the process is known as vector quantization. We shall often abbreviate vector quantization here as VQ.

Vector quantization is presented as a process of redundancy removal that makes effective use of four interrelated properties of vector parameters: linear dependency, nonlinear dependency, shape of the probability density function, and vector dimensionality itself. VQ technique is very powerful and convenient, and has been applied in various areas. For example, VQ in speech coding has reduced the transmission rate of 2400-bit/s vocoders so that they can operate at much lower rates while maintaining acceptable speech intelligibility and quality (see, for example, [Buso, 1980] [Juang, 1982]). Speech coding at very low rates, in the range of 200-800 bit/s, has attracted substantial interest for use in commercial applications (see, for example, [Rocus, 1982] [Wong, 1983]). VQ has also been used regularly and effectively in pattern-recognition types of speech applications, such as for speech and speaker recognition (see, for example, [Levinson, 1985][Rabiner, 1983]). The VQ problem is, after all, part of the general pattern-recognition problem of how to classify data into a discrete number of categories that optimize some fidelity criterion.

2.1 A Problem Formulation

We assume that $\mathbf{x}=[x_1\ x_2\ \ldots\ x_N]^T$ is an N-dimensional vector whose components $\{x_k,\ 1 \leqq k \leqq N\}$ are real-valued, continuous-amplitude random

variables. (The superscript T denotes transpose.) In vector quantization, the vector x is mapped onto another real-valued, discrete-amplitude, N-dimensional vector y. We say that x is quantized as y, and y is the quantized value of x. We write

$$\mathrm{y} = q(\mathrm{x}) \qquad (2.1)$$

where $q(.)$ is the quantization operator. The vector y is also called the output vector corresponding to x. Typically, y takes on one of a finite set of values $\mathrm{Y} = \{ \mathrm{y}_i, 1 \leqq i \leqq L \}$, where $\mathrm{y}_i = [\, y_{i1}\ y_{i2}\ \ldots\ y_{iN} \,]^T$. The set of Y is referred to as the codebook, L is the size of codebook, and $\{ \mathrm{y}_i \}$ is the set of code vectors. The vectors y_i are also known in pattern-recognition literature as the reference patterns or templates. The size L of the codebook is also called the number of levels, a term borrowed from scalar quantization terminology. Thus, one talks about an L-level codebook or L-level quantizer. To design such a codebook, we partition the N-dimensional space of the random vector x into L regions or cells $\{ C_i, 1 \leqq i \leqq L \}$ and associate with each cell C_i a vector y_i. If x is in C_i, the quantizer then assigns the code vector y_i ; that is,

$$q(\mathrm{x}) = \mathrm{y}_i, \qquad if\ x \in C_i. \qquad (2.2)$$

This codebook design process is also known as training.

2.2 A Distortion Measure

When x is quantized as y, a quantization error results and a distortion measure $d(\mathrm{x},\mathrm{y})$ can be defined between x and y. In this paper we use a measure called WLR (Weighted Likelihood Ratio), which is based on LPC analysis [Sugiyama, 1981]. WLR is defined so as to emphasize spectral peaks, because human auditory system is more sensitive to the mountain-shaped portion, such as formant frequencies, than to the valley-shaped portion of the sound spectrum [Matsuda, 1966] [Flanagan, 1972].

The spectrum obtained through LPC analysis can be represented by an all-pole function

$$f(\lambda) = \frac{u \cdot R_f}{\left| \sum_{n=0}^{p} a_n e^{jn\lambda} \right|^2} \qquad (2.3)$$

$$g(\lambda) = \frac{v \cdot R_g}{\left| \sum_{n=0}^{p} b_n e^{jn\lambda} \right|^2} \qquad (2.4)$$

where f and g are the respective LPC spectra of reference and input patterns; a_n, R_f, u and b_n, R_g, v are prediction coefficients, predicted normalized residuals and powers of f and g in LPC analysis, respectively; p is the order of analysis and λ is the angular frequency. The WLR measure is defined as follows:

$$WLR = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left\{ \left(log \frac{f}{g} + \frac{g}{f} - 1 \right) \cdot \frac{f}{u} + \left(log \frac{g}{f} + \frac{f}{g} - 1 \right) \cdot \frac{g}{v} \right\} d\lambda \qquad (2.5)$$

In Eq.(2.5), $log(f/g) + (g/f) - 1$, $log(g/f) + (f/g) - 1$ are terms indicating the differences of spectra, and f/u and g/v are terms representing the weight of peaks.

If the power ratio u/v is set so that the measured value is the minimum, Eq.(2.5) can easily be calculated as follows using the parameters of LPC analysis:

$$WLR = \sum_{n=1}^{N} (r_n - r_n')(c_n - c_n') \qquad (2.6)$$

where r_n, r_n' and c_n, c_n' are correlation coefficients and LPC cepstrum coefficients of f and g, respectively.

2.3 Codebook Design

As mentioned above, to design an L-level codebook, we partition N-dimensional space into L cells $\{C_i, 1 \leqq i \leqq L\}$ and associate a vector $\mathbf{y}_i$ with each cell C_i. The quantizer then assigns the code vector $\mathbf{y}_i$ if $\mathbf{x}$ is in C_i. A quantizer is said to be an optimal (minimum-distortion) quantizer if the distortion is

minimized over all L-level quantizers. There are two necessary conditions for optimality. The first condition is that the optimal quantizer is realized by using a minimum-distortion or nearest neighbor selection rule

$$q(\mathbf{x})=\mathbf{y}_i, \ iff\, d(\mathbf{x},\mathbf{y}_i) < d(\mathbf{x},\mathbf{y}_j), \ j \neq i, \ 1 \leqq j \leqq L. \qquad (2.7)$$

That is, the quantizer chooses the code vector that results in the minimum distortion with respect to x. The second condition for optimality is that each code vector $\mathbf{y}_i$ is chosen to minimize the average distortion in cell C_i. That is, $\mathbf{y}_i$ is that vector y which minimizes

$$D_i = \int_{x \in C_i} d(x,y)\, p(x)\, dx\,. \qquad (2.8)$$

We call such a vector the centroid of the cell C_i, and we write

$$\mathbf{y}_i = cent(C_i). \qquad (2.9)$$

Computing the centroid for a particular region will depend on the definition of the distortion measure. In practice, we are given a set of training vectors $\{\mathbf{x}(n), 1 \leqq n \leqq M\}$. A subset M_i of those vector will be in cell C_i. The average distortion D_i is then given by

$$D_i = \frac{1}{M_i} \sum_{x \in C_i} d(x, y_i) \qquad (2.10)$$

For either the mean square error or the weighted mean square error criterion, one can show that D_i is minimized by

$$y_i = \frac{1}{M_i} \sum_{x \in C_i} x(n)\,, \qquad (2.11)$$

or $\mathbf{y}_i$ is simply the sample mean of all the training vectors contained in C_i. One method for codebook design is an iterative clustering algorithm known in pattern-recognition literature as the K-means algorithm. In our problem here, $K=L$. This algorithm divides the set of training vectors $\{\mathbf{x}(n)\}$ into L clusters C_i in such a way that the two necessary conditions for optimality are satisfied. Below,

m is the iteration index and $C_i(m)$ is the i-th cluster at iteration m, with $y_i(m)$ its centroid. The algorithm is as follows:

Step 1: Initialization: Set $m=0$. Choose, by an adequate method, a set of initial code vectors $y_i(0), 1 \leqq i \leqq L$.

Step 2: Classification: Classify the set of training vectors $\{x(n), 1 \leqq n \leqq M\}$ into the clusters C_i by the nearest neighbor rule.

$$x \in C_i(m), \text{ iff } d[x, y_i(m)] \leqq d[x, y_j(m)], \text{ for all } j \neq i.$$

Step 3: Code Vector Updating: $m \leftarrow m+1$. Update the code vector of every cluster by computing the centroid of the training vectors in each cluster

$$y_i(m) = cent(C_i(m)), 1 \leqq i \leqq L.$$

Step 4: Termination Test: If the decrease in overall distortion $D(m)$ at iteration m relative to $D(m-1)$ is below a certain threshold, stop; otherwise go to Step 2.

The above algorithm can be shown to converge to a local optimum. Furthermore, any such solution is, in general, not unique. Global optimality may be approximately achieved by initializing the code vectors to different values and repeating the above algorithm for several sets of initializations and then choosing the codebook that results in the minimum overall distortion.

3. Voice Conversion based on Codebook Mapping

The voice conversion algorithm consists of two steps: a learning step and a conversion-synthesis step. The learning step generates the mapping codebooks, and the conversion-synthesis step uses them to synthesize.

3.1 Learning Step

The mapping codebooks describe a mapping function between the vector spaces of two speakers. Figure 2.2 shows a block diagram of the procedure for generating a mapping codebook for spectrum parameters.

Step 1: Speakers, A and B, pronounce a learning word set used to generate a codebook for each speaker. Then, learning words uttered by speaker A are vector quantized using his/her codebook. The same words uttered by speaker B are also vector quantized in the same way.

Step 2: The correspondence between the vectors of the same words from the two speakers is determined by using the Dynamic Time Warping (DTW).

Step 3: The vector correspondences between the two speakers are accumulated as histograms for all learning words.

Step 4: Using the histogram for each code vector of speaker A as a weighting function, a mapping codebook from speaker A to B is defined as a linear combination of speaker B's vectors.

Step 5: Steps 2, 3, and 4 are repeated to refine the mapping codebook.

Pitch frequencies and power values contribute heavily to speech individuality. Mapping codebooks for these parameters are also generated at the same time using almost the same procedure mentioned above, with these differences:

1. pitch frequencies and power values are each scalar-quantized, and
2. the mapping codebook for pitch frequencies is defined based on the maximum occurrence in the histogram.

3.2 Conversion-Synthesis Step

Figure 2.3 shows a block diagram of the conversion-synthesis step. First, speaker A's speech is analyzed by the linear prediction method. Then the spectrum parameters are vector-quantized using his/her codebook, and parameters for pitch frequencies and power values are scalar-quantized using his/her codebooks. Next, all parameters are decoded using the speaker A to B mapping codebooks between speakers A and B. Finally, speech is synthesized by an LPC vocoder. The output speech will have the voice individuality of speaker B.

4. Performance Evaluation

4.1 Evaluation by Distortion

To evaluate the performance of this conversion technique, we measured the distortion of the spectrum parameters as well as of the pitch frequencies.

4.1.1 Spectrum conversion experiments

Experiment conditions are listed in Table 2.1. A set of 100 phonetically-balanced learning words was used to produce mapping codebooks. Spectrum conversions were made between female and male voices, between male and male, and between female and female voices. Six speakers (3 male and 3 female speakers, all professional announcers) provided speech material.

Table 2.2 lists the results of the open test. After vector-quantization, two kinds of spectrum distortions between two speech samples were calculated: between the input and target speaker's ("before conversion"), and the converted and target speaker's speech ("after conversion"). For the female-to-female conversion, the distortion decreased by 27% compared to nonconversion, for the male-to-male conversion by 49%, and for the male-to-female conversion by 66%. These results

show that this conversion technique is highly effective when there is a large enough difference between two speakers's voices.

4.1.2 Pitch frequency conversion experiments

Pitch frequency was also converted using the same process described in 4.1.1, and the experiment results are shown in Fig. 2.4. This figure shows the relationship between the number of learning words and the average pitch frequency differences after conversion. The value at 0 learning word shows the natural average pitch frequency difference between the two speakers. Regardless of speaker combinations, 60 words are enough to make a mapping codebook for pitch frequency that reduces the average pitch frequency difference to less than 15 Hz.

4.2 Evaluation by Listening Test

To evaluate the overall performance of this technique, three kinds of listening tests were carried out. The first dealt with male-to-female conversion and the other two with male-to-male conversion.

4.2.1 Experiment procedure

4.2.1.1 Experiment 1

Experiment 1 was designed to evaluate voice quality for male-to-female voice conversion by a pair-comparison listening test. In addition to the fully converted speech, pitch and spectrum parameters were also converted separately in order to examine their individual contributions to speech individuality. The following is a list of 5 different speech conversions performed in this experiment.

1. vector-quantized original male speech (m)

2. male-to-female converted speech: pitch frequency conversion only (mp-fp)

3. male-to-female converted speech: spectrum conversion only (ms-fs)

4. male-to-female converted speech: all parameters (m-f)

5. vector-quantized original female speech that is the target for the conversions (f)

To avoid unnecessary cues for the judgment of voice quality, different words were used to make speech pairs for the listening test. A set of speech pairs consisted of all possible stimuli combinations from the 5 different conversions, 40 in total. They were presented to listeners through a loud-speaker in a sound-proof room. Twelve listeners were asked to rate the similarity of each pair into five categories: "similar", "slightly similar", "difficult to decide", "slightly dissimilar", "dissimilar".

4.2.1.2 Experiment 2

Experiment 2 was designed to evaluate the conversion between two male speakers by the so-called ABX method. Stimuli A and B are vector-quantized original speech tokens for speakers A and B. The stimulus X takes either the converted token (A→B or B→A) or the vector-quantized original token (A or B). Four different words were used for the conversions and each triad was a combination of 3 different words. A total of 96 speech triads were presented to the listeners. The listeners were required to select the stimulus (A or B) more closely resembling the stimulus X.

4.2.1.3 Experiment 3

Experiment 3 was designed to evaluate the conversion between male speakers in the same way as in 4.2.1.1. Conversions for pitch frequencies alone and spectrum parameters alone, however, were excluded. The following is a list of the 4 conversions used.

1. vector-quantized male speech (male 1)

2. same as 1 but for another male speaker (male 2)

3. converted speech from male 1 to male 2 (male 1→male 2)

4. converted speech from male 2 to male 1 (male 2→male 1)

A total of 72 speech pairs were generated using the same procedures as in Experiment 1.

4.2.2 Experiment results

4.2.2.1 Evaluation of male-to-female conversion (Experiment 1)

Hayashi's fourth method of quantification[Hayashi, 1985] was applied to the experimental data obtained by the listening test. This method places stimuli in a space according to the similarities between every two stimuli. Its formulation minimizes the measure Q,

$$Q = -\sum_{i,j} e(i,j)\{x(i) - y(j)\}^2 \qquad (2.12)$$

where $e(i,j)$ denotes the similarity between stimuli i and j, and $x(i)$, $y(j)$ represent the locations of stimulus i in the space.

The projection onto a two-dimensional space is shown in Fig. 2.5. This figure shows the relative similarity-distance between stimuli. "m→f" converted speech is very close to the speech "f", indicating that this technique properly converted the male speech to the target female speech. Judging from the positions of "mp→fp" and "ms→fs", we see that the first and second axes roughly correspond to pitch frequency and spectrum differences, respectively. This indicates that neither pitch frequency nor spectrum carries enough information about speech individuality, and that both are necessary.

4.2.2.2 Evaluation of male-to-male conversion (Experiments 2 and 3)

The results of Experiment 2 are listed in Table 2.3. The numbers in this table are the percentages of responses in which stimuli X was judged correctly. These results show that listeners can't always correctly identify the speaker, even if the original speaker's speech is used as stimuli X; i.e., the correct answer is about

85% in the male 1, male 2 pair, and about 60% in the male 1, male 3 pair. Judging from these facts, the conversion between male 1, male 2 was satisfactorily performed.

The relatively poor performance for the male 1-male 3 conversion stems from the fact that male 1's voice quality is very similar to male 3's voice. This similarity can be seen by the small "distance" between the original speaker and target speaker shown in Fig. 2.6. This figure shows spectrum distance and pitch frequency distance before and after conversion.

Figure 2.7. shows the results for Experiment 3 analyzed by the same method as in 4.2.2.1. The converted speech samples, "male 1→male 2" and "male 2→male 1", are both placed close to their target speech. This indicates that the proposed technique can also convert speech individuality between speakers of the same-sex.

5. Improved Voice Conversion Algorithms

As discussed in section 4, voice conversion based on codebook mapping successfully changes speech individuality. In other words, voice conversion can be well formulated as a mapping problem between speakers' codebooks. However, VQ also introduces quantization errors, which result in losing naturalness and clarity of synthesized speech. In this section, we propose new algorithms to improve the quality of converted speech itself and evaluate the performance.

5.1 Proposed Algorithms

On general principle, the larger the size of the codebook is, the smaller the quantization errors are. However, there are the following problems on increase of the codebook size:

1. To optimally design a large codebook and a mapping codebook, we need not only a large amount of data, but also expensive computational costs. In the K-means algorithm, most of the computation result from the classification step. For an L-level quantizer, M training vectors, and l iterations, the computational cost for training is

$$NLMl = N2^{NR}Ml. \qquad (2.13)$$

   For reliable design of the codebook, one needs at least 10 and preferably about 50 training vectors per code vector, so that M is on the order of 10L or more.

2. Although VQ is very powerful and efficient in less than 10 bits speech coding, the efficiency is saturated more than 10 bits[Moriya, 1982].

Therefore we take another strategy to improve the quality of synthesized speech; i.e., the usage of information in input speech as much as possible. In the strategy, we have a hypothesis that the most important mapping rules required in speech individuality conversion have achieved by codebook mapping.

5.1.1 Improvement using fuzzy VQ

Fuzzy VQ is one technique to approximate an input vector by linear combination of code vectors[Ruspini, 1970]. Therefore Fuzzy VQ can represent various kinds of vectors beyond limitation caused by the codebook size, and approximate input vectors more precisely than conventional VQ. Fuzzy VQ is defined as follows:

$$u_i = \frac{1}{\sum_{j=1}^{k} \left(\frac{d_i}{d_j} \right)^{\frac{1}{(m-1)}}} \qquad (2.14)$$

$$X' = \frac{\sum_{i=1}^{k} \left[\left(u_i\right)^m \cdot V_i \right]}{\sum_{i=1}^{k} \left(u_i\right)^m}$$

$$= \sum_{i=1}^{k} \left(\frac{\left(u_i\right)^m}{\sum_{i=1}^{k} \left(u_i\right)^m} \right) \cdot V_i$$

$$= \sum_{i=1}^{k} u_i' \cdot V_i \qquad (2.15)$$

where u_i; fuzzy membership function. $u_i \in [0,1]$ $^{\forall} i$

$d_i = \| X - V_i \|$. V_i is a code vector in codebook V.

m: fuzziness.

X': decoded spectrum of input spectrum X.

k: number of code vectors.

The proposed algorithm using Fuzzy VQ is as follows:

1. A mapping codebook is generated between speaker A and speaker B.
2. In the conversion-synthesis step, input spectrum parameters X are fuzzy vector quantized, where k-th nearest code vectors to input spectrum are used(k = 6).

$$X' = \sum_{i=1}^{k} u'_i V_i \qquad (2.16)$$

where V_i; a code vector in speaker A's codebook V.

X': a fuzzy vector quantized vector.

3. In local spectrum space, it could be considered that the spectrum space has smooth shape. A mapping codebook provides discrete (code vector) correspondence of spectrum space between different speakers. Therefore if an input vector(spectrum) is represented using k-th nearest code vectors by fuzzy VQ, it is reasonable hypothesis that fuzzy membership function

taken from speaker A's spectrum space is preserved in speaker B's spectrum space (refer Fig.2.8). After all, conversion is performed by replacing speaker A's code vectors by mapping code vectors as follows:

$$X^{map} = \sum_{i=1}^{k} u'_i V_i^{map} \qquad (2.17)$$

where V_i^{map}; a code vector in mapping codebook V^{map}.

X^{map}: a converted vector.

5.1.2 Improvement using difference vector

In addition to applying fuzzy VQ, an input vector is modified using difference vectors between the input vector and code vectors. The usage of difference vectors makes it possible to represent various spectra beyond the limitation caused by the codebook size. The algorithm is as follows:

1. A mapping codebook is generated between speaker A and speaker B.
2. Basically conversion is performed by the following equation:

$$\begin{aligned} X^{map} &= \left(X - V_i\right) + V_i^{map} \\ &= \left(V_i^{map} - V_i\right) + X \quad (i=1,2,....n) \end{aligned} \qquad (2.18)$$

where V_i; a code vector in speaker A's codebook V.

V_i^{map}: a code vector in a mapping codebook V^{map}.

X: an input vector.

X^{map}: a converted vector.

3. In the conversion-synthesis step, input spectrum parameters X are fuzzy vector quantized by Eq. (2.15). Using the fuzzy membership function u' as a weighting function, conversion is finally performed by the following equation(refer Fig.2.9):

$$X^{map} = \sum_{i=1}^{k} u'_i \left(V_i^{map} - V_i\right) + X \quad (i=1,2,....k) \qquad (2.19)$$

In the following evaluation experiments, difference vectors are calculated between LPC spectrum envelopes. LPC spectrum envelope is sampled at 256 evenly spaced points. Once add operation is applied to LPC spectrum envelopes, the modified envelope can not be converted back to LPC coefficients. Therefore after transforming LPC spectrum envelopes to waveforms by inverse Fourier transform, speech is synthesized by overlap adding method. Finally, the synthesized speech is again analyzed by LPC method then output speech is synthesized by LPC. A block diagram of the algorithm using difference vectors is shown in Fig. 2.10.

5.2 Evaluation by Listening Test

To evaluate the performance of the improved algorithms, pair-comparison listening tests are carried out among a basic algorithm that was explained in section 3, the improved algorithm using fuzzy VQ and the improved algorithm using difference vectors. Ten words are synthesized using the three algorithms and all combinations of synthesized speech are presented to 12 listeners, and they are asked to indicate the better one.

Table 2.4 shows the experiment results. Judging from the table, the both improved algorithms have significantly better performance than the basic algorithm, and the improved algorithm using difference vectors has the best performance. Main improvements of the synthesized speech are summarized in two points; i.e., improvements in smoothness and clarity improvements in consonants.

Improvements in smoothness means reduction of click and rough noise, which result in reducing artificial sound and increasing naturalness of the synthesized speech. This effect can be obtained by both the usage of fuzzy VQ and the usage of difference vectors. Figure 2.11 shows speech waveform and LPC spectrum envelope of converted speech by VQ(the basic algorithm) and fuzzy VQ (the improved algorithm). As shown in the figure, both waveform and LPC spectrum change smoothly when fuzzy VQ is applied. That means fuzzy VQ makes it

possible to generate more spectrum patterns beyond the limitation by the codebook size and to successfully interpolate code vectors.

Clarity improvements in consonants can be obtained by only the usage of difference vectors. In Fig. 2.12, LPC spectrum envelope of original speech, target speech, converted speech using fuzzy VQ, converted speech using difference vectors are shown and also difference vectors are shown. According to the amplitude of the difference vectors, the speech segment in the figure is divided into two parts; i.e., part (A) where difference vectors are almost equal to zero, and part (B) where difference vector has large amplitude. An analysis of the result of the listening test reveals that the clarity in the part (A) is improved and the part (A) is correspond to consonant region. This improvement can be also shown in part (A) of Fig. 2.12. The second lowest peak of LPC spectrum can be observed in the converted speech using difference vectors as well as in both the original speech and the target speech synthesized speech. Judging from these results, the usage of difference vectors is very effective to represent details in spectrum characteristics that are ignored by VQ or fuzzy VQ.

6. Conclusion

In this chapter, new voice conversion algorithms based on codebook mapping were proposed. The advantage of this technique are summarized as follows:

1. The mapping codebooks which make it possible to give an individuality to synthesized speech are generated from a limited number of word utterances.

2. The mapping codebooks enable voice conversion of high quality between any two speakers.

3. The synthesis process requires few computation and produces speech in real time.

The performance of this technique is confirmed by spectrum distortion and pitch frequency difference. The spectrum distortion between original speech and

target speech decreased by a range of 27% to 66%. Pitch frequency difference decreased to less than 15Hz. The overall performance of this technique is also confirmed by listening tests. It can be concluded that the converted speech has a voice quality very close to the target speaker's.

To improve naturalness and clarity of the converted speech, the usage of fuzzy VQ and difference vectors was discussed. According to listening test, fuzzy VQ makes it possible to improve smoothness by generating more spectrum patterns beyond the limitation by the codebook size and the usage of difference vectors is very effective to improve clarity by representing details in spectrum characteristics that are ignored by VQ or fuzzy VQ.

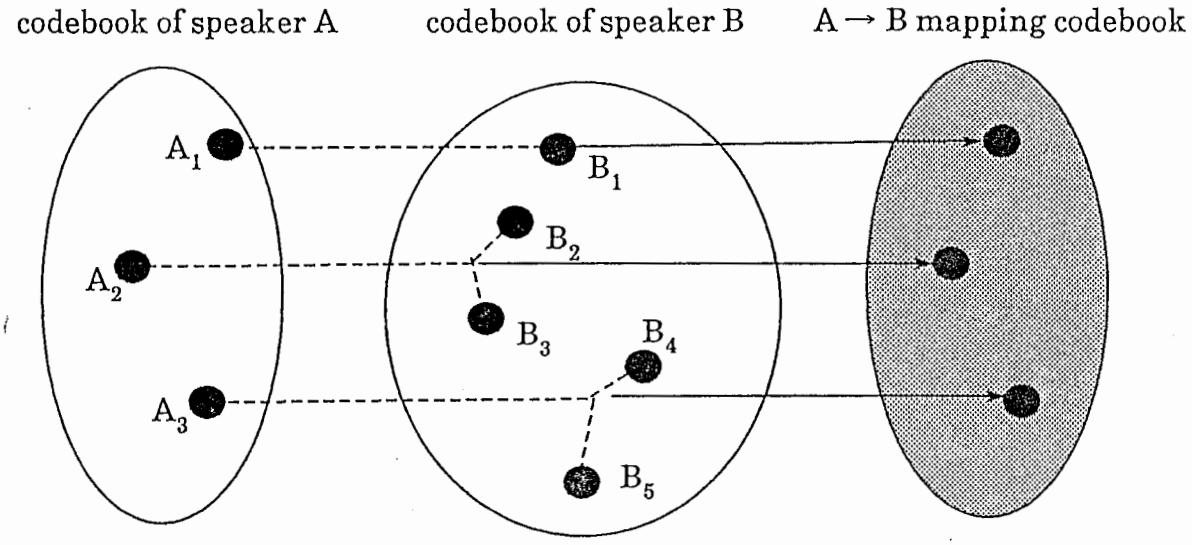


Fig.2.1 Basic idea of voice conversion

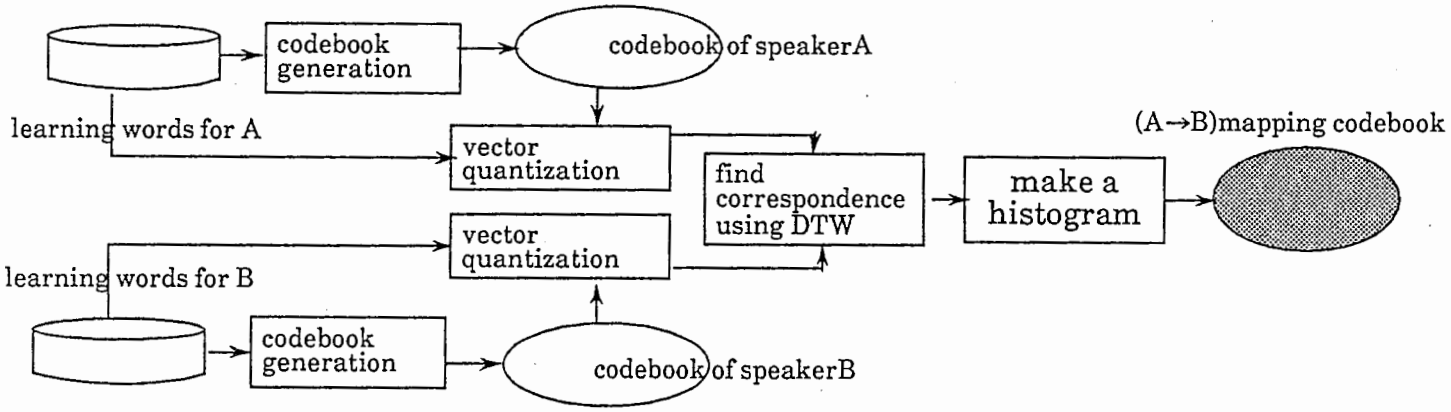


Fig.2.2. Method for generating a mapping codebook

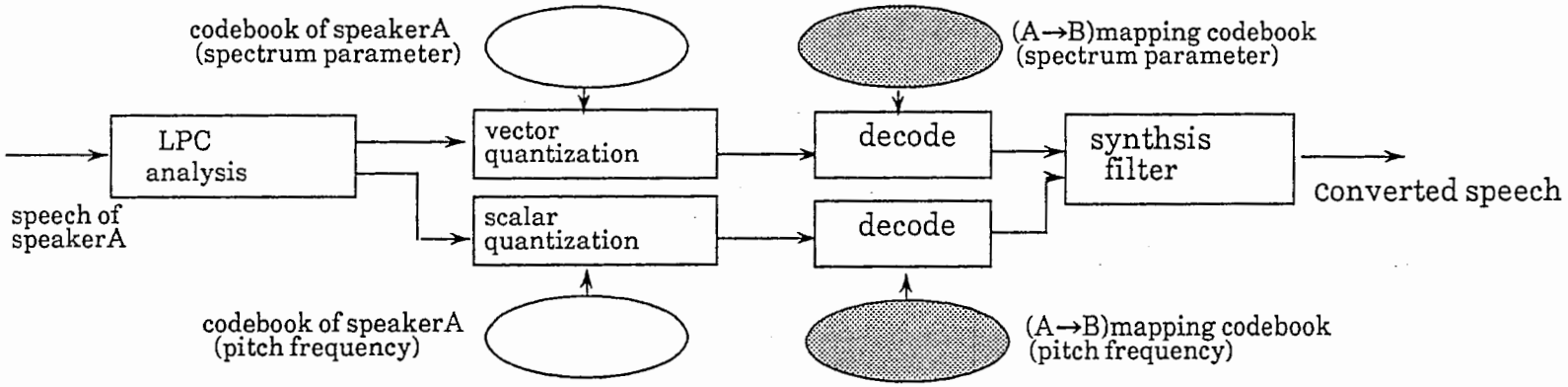


Fig.2.3. Block diagram of voice conversion from speaker A to speaker B

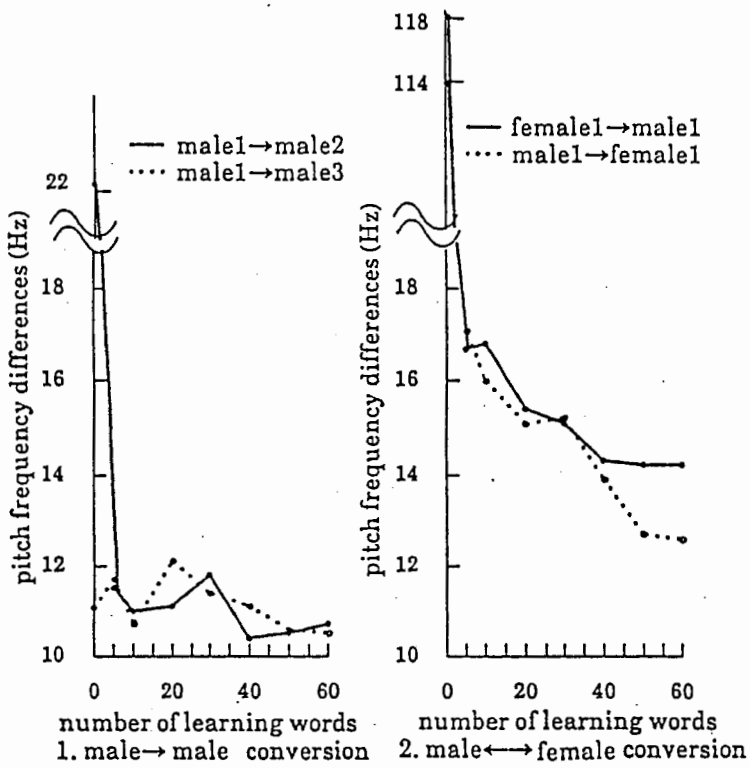
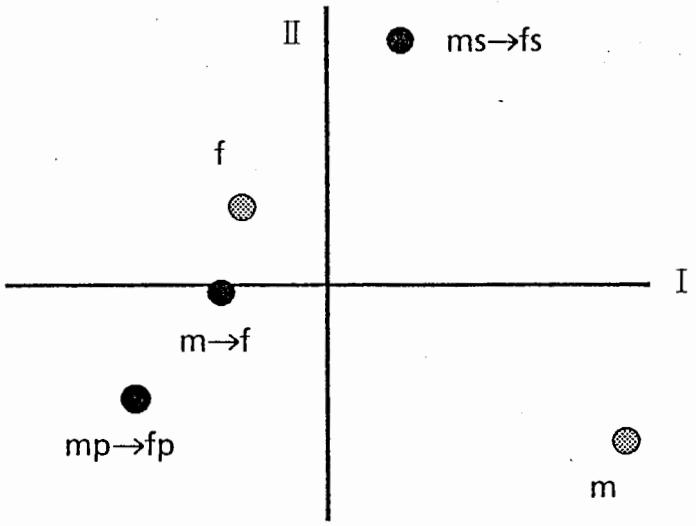


Fig.2.4. Pitch frequency differences for number of learning words



m: vector-quantized original male speech
f: vector-quantized original female speech
mp→fp: male-to-female converted speech
pitch frequency only
ms→fs: male-to-female converted speech
spectrum parameters only
m→f: male-to-female converted speech
all parameters

Fig.2.5. Distribution of psychological distances for the male-to-female voice conversion

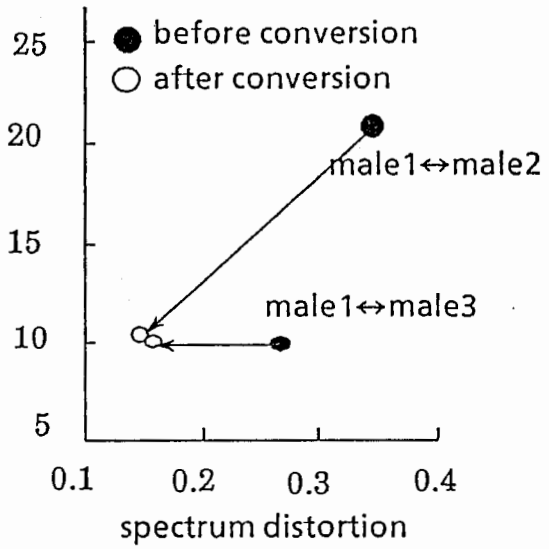
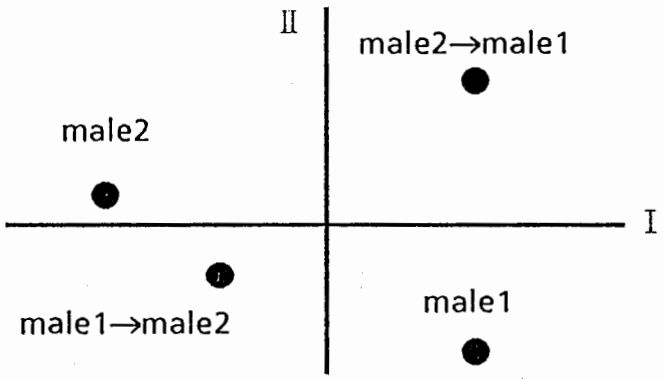


Fig.2.6. Speaker distance in spectrum distortion and pitch frequency difference



male1: vector-quantized male 1 speech
male2: vector-quantized male 2 speech
male1→male2: converted speech from male 1 to male 2
male2→male1: converted speech from male 2 to male 1

Fig.2.7. Distribution of psychological distances for the male-to-male voice conversion

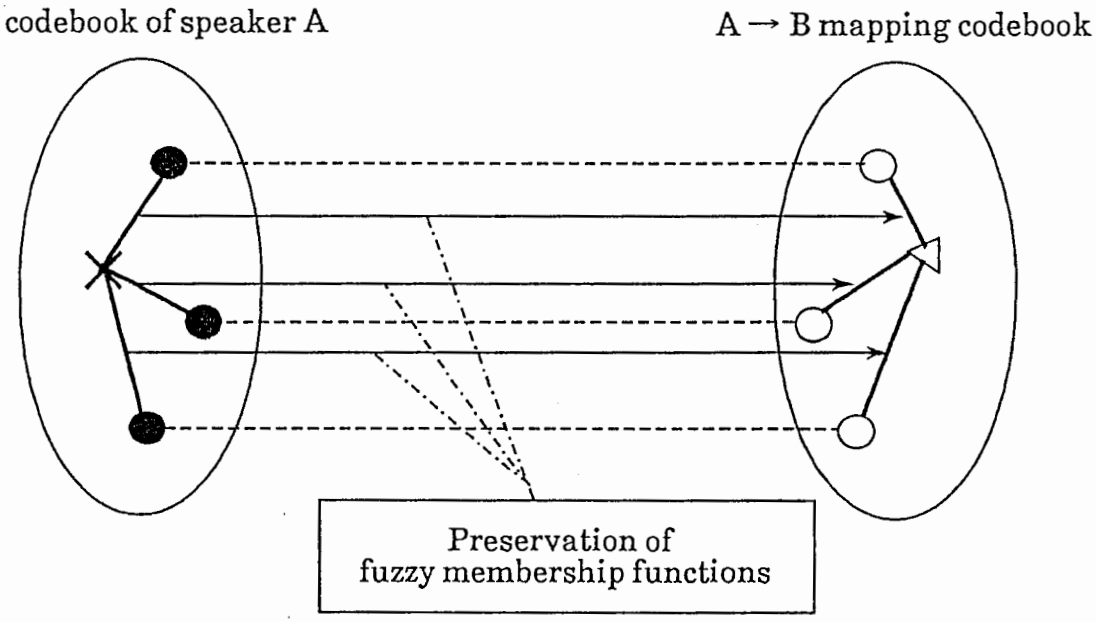


FIg. 2.8 Basic idea of voice conversion by fuzzy VQ

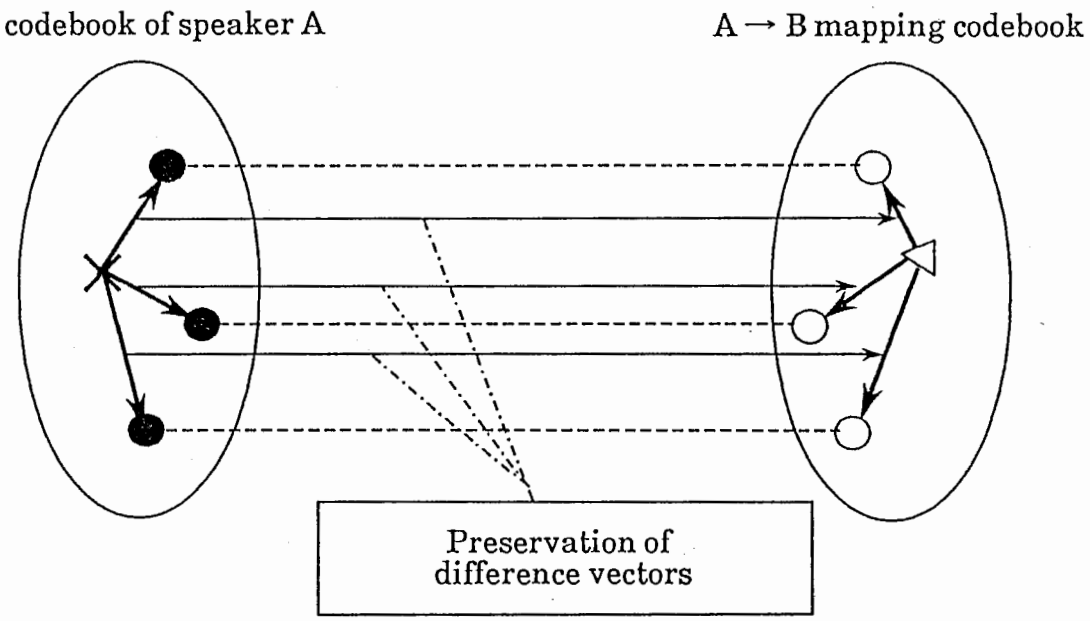


FIg. 2.9 Basic idea of voice conversion by difference vectors

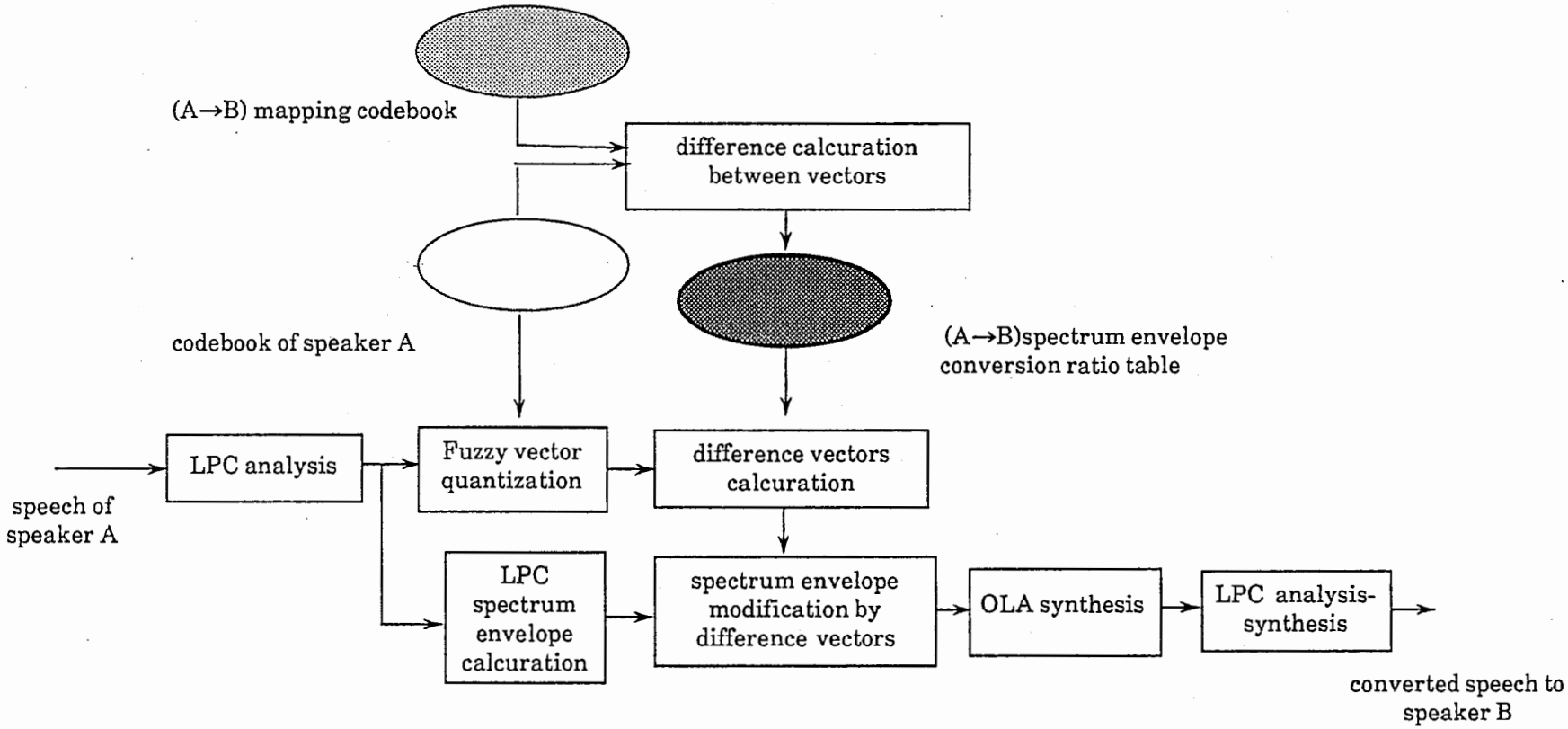
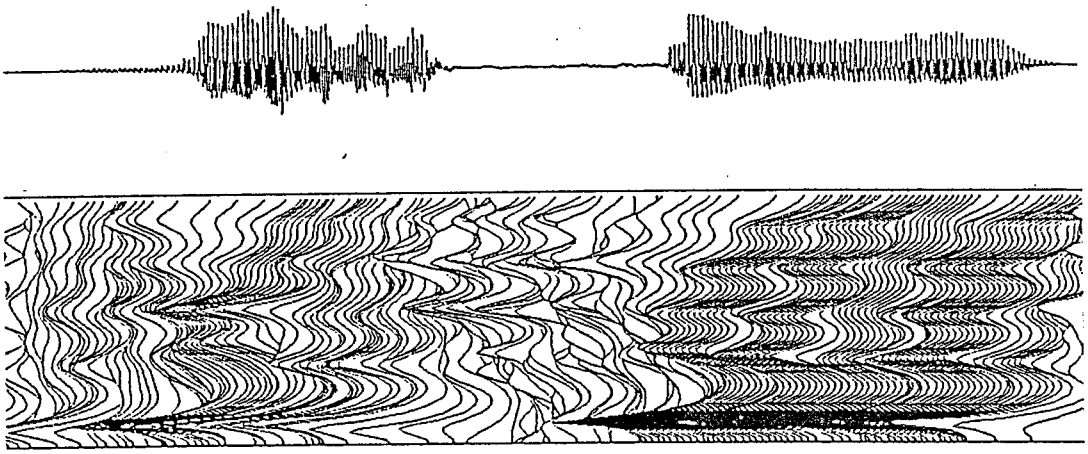
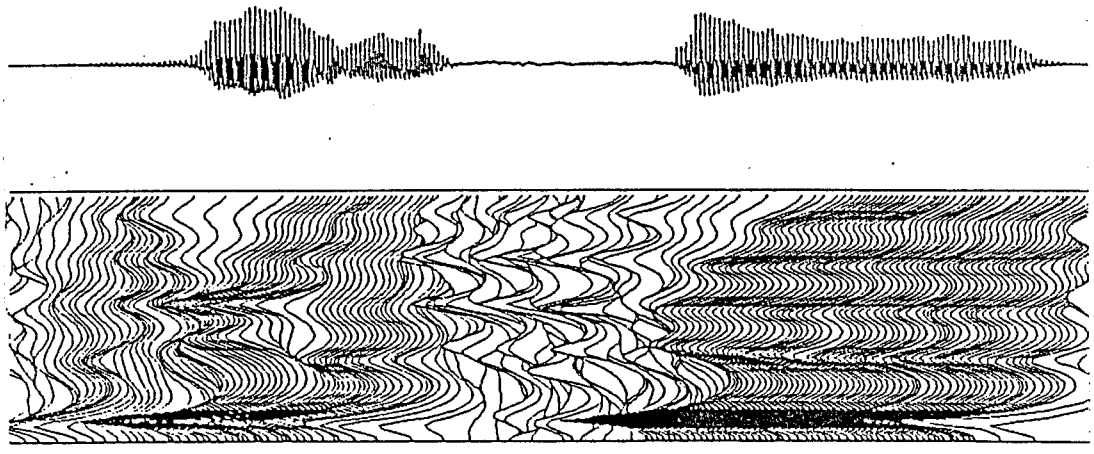


Fig.2.10 Block diagram of a conversion from speaker A to speaker B by difference vectors



(a) Vector quantization



(b) Fuzzy vector quantization

Fig.2.11. Speech waveforms and LPC running spectra of converted speech

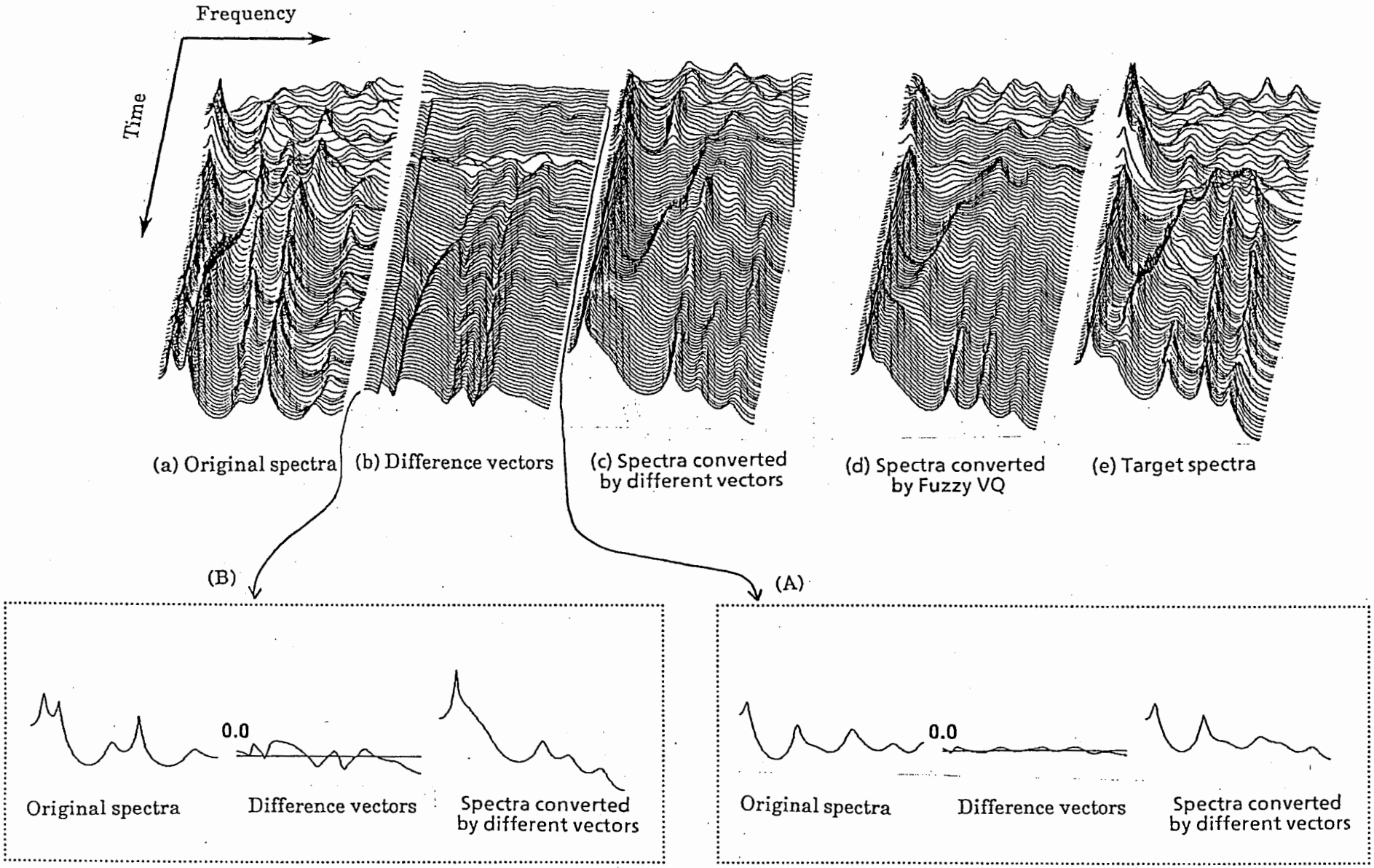


Fig.2.12. LPC running spectra of converted speech

Table 2.1 Experiment conditions

A/Ddata	12KHz sampling、16bit
window length	256points (21.3msec)
window shift	36points (3.0msec)
analysis order	12
clustering measure	WLR(Weighted Likelihood Ratio)
learning samples for clustering	5,000 frames
codebook size for spectrum parameter	256
learning words for mapping	100
codebook size for pitch frequency	35 ~ 64

Table 2.2 Spectrum distortion

speaker combination	before conversion	after conversion
female1→female2	0.2759	0.2109
female1→female3	0.2070	0.1489
male1→male2	0.3364	0.1717
male1→male3	0.2851	0.1550
male1→female1	0.6084	0.2193

Table 2.3 Percentage of correct responses

speaker combination	correct response(%)	speaker	correct response(%)
male 1→male 2	69.4	male 2	84.7
male 2→male 1	75.0	male 1	85.4
male 1→male 3	46.5	male 3	64.6
male 3→male 1	38.8	male 1	58.3

Table 2.4 Preference score

Stimuli combinations (better > worse)	Preference score (%)
Fuzzy VQ > VQ	73.0
Difference vector > VQ	76.0
Difference vector > Fuzzy VQ	62.9

Chapter 3

Pitch Modification by Signal Reconstruction

1. Introduction

Speech modification algorithms play an important role in speech applications. For example, modification of pitch frequency and duration is necessary in dyad-based synthesis-by-rule systems, and the spectrum envelope should also be modified in a voice quality control system. Although the modification of such parameters is possible using conventional vocoder algorithms, the modified speech quality is not enough, especially for pitch frequency modification. In this chapter we propose a new algorithm which makes it possible to synthesize high quality speech even if pitch frequency or duration is somewhat changed.

To achieve high quality, the proposed algorithm is developed based on the Short-Time Fourier Transform(STFT) synthesis. The synthesis algorithm can theoretically reproduce the original speech from analysis parameters. Therefore, all that is necessary to achieve high quality modified speech is to modify these parameters appropriately. Some speech modification algorithms based on the STFT have been proposed[Portnoff, 1981][Seneff, 1982][Roucos, 1985][Charpentier, 1986]. The following two points are new technical issues in the proposed algorithm. First, an algorithm is adopted to separate spectrum envelope and source components from the speech signal. The parameters which contribute only to a desired modification should be changed. From this point of view, a new cepstrum lifter is proposed. Second, a new algorithm to control phase spectrum is proposed. In the modification algorithms based on the STFT that have been proposed, the phase unwrapping and the phase control are not only very important but also very complex procedures. The proposed algorithm eliminates these problems by introducing window shift control and the signal reconstruction algorithm[Griffin, 1984].

In Section 2, STFT analysis and synthesis are briefly reviewed. In the Section 3, the details of the proposed algorithm are explained. In Section 4, the performance of the algorithm is evaluated through a listening test.

2. Short-Time Fourier Transform of a Sequence

In this section, we define the STFT representation for a sequence[Lim, 1988]. A major theme used throughout this section is that the representation for the STFT of a sequence is analogous to the Fourier transform representation of a sequence.

2.1 Fourier Transform View

The STFT is presented as an extension to the basic Fourier transform definitions for a sequence. In particular, we introduce the discrete-time STFT and the discrete STFT as counterparts to the discrete-time Fourier transform and the discrete Fourier transform, respectively. The discrete-time STFT is related to the discrete-time Fourier transform, which is given by

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad (3.1)$$

where ω is a continuous variable denoting frequency. The discrete-time STFT of $x(n)$ is a set of such discrete-time Fourier transforms corresponding to different time sections of $x(n)$. The time section for time n_0 is obtained by multiplying $x(n)$ with a shifted sequence $w(n_0-n)$. The expression for the discrete-time STFT at time n_0 is therefore given by

$$X(n_0, \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n_0-n)\, e^{-j\omega n} \qquad (3.2)$$

where $w(n)$ is referred to as the analysis window. The sequence $f_{n_0}(n)=x(n)w(n_0-n)$ is generally called a short-time section of $x(n)$ at time n_0. This sequence is obtained by time-reversing the analysis window $w(n)$, shifting the result by n_0 points. and multiplying it with $x(n)$. Once we have the short-time section for time n_0, we can take its Fourier transform to obtain the frequency function $X(n_0, \omega)$ with n_0 fixed. To obtain $X(n_0+1, \omega)$, we slide the time-reversed analysis window one point from its previous position, multiply it with $x(n)$, and take the Fourier

transform of the resulting short-time section. Continuing this way, we generate a set of discrete-time Fourier transforms that together constitute the discrete-time STFT. We obtain the mathematical representation for the STFT by replacing the fixed n_0 of Eq.(3.2) as m. We thus obtain the STFT definition:

$$X(n, \omega) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)\ e^{-j\omega m} \qquad (3.3)$$

For digital processing, we use the discrete STFT, which is related to the discrete-time STFT in the same manner as the DFT is related to the discrete-time Fourier transform. Recall that the DFT $X(k)$ of a finite-duration sequence $x(n)$ is obtained by sampling the discrete-time Fourier transform over one period. That is,

$$X(k) = X(\omega)|_{\omega=2\pi/N} R_N(k) \qquad (3.4)$$

where N is the frequency sampling factor and $R_N(k)$ is an N-point rectangular sequence given by

$$R_N(k) = u(k) - u(k-N) \qquad (3.5)$$

In analogy, the discrete STFT is obtained from the discrete-time STFT through the following relation:

$$X(n, k) = X(n, \omega)|_{\omega=2\pi/N} R_N(k) \qquad (3.6)$$

where we have sampled the discrete-time STFT with a frequency sampling interval of $2\pi/N$ to obtain the discrete STFT. Substituting Eq.(3.3) into Eq.(3.6), we obtained the following relation between the discrete STFT and its corresponding sequence $x(n)$:

$$X(n, k) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)\ e^{-j2\pi km/N} R_N(k) \qquad (3.7)$$

In many applications, the time variation (the n dimension) of $X(n,k)$ is decimated by a temporal decimation factor L to yield the function $X(nL,k)$.

Just as the discrete-time STFT can be viewed as a set of Fourier transforms of the short-time sections $f_n(m)$, the discrete STFT in Eq.(3.7) is easily seen to be a set of DFTs of the short-time sections $f_n(m)$. When the time dimension of the discrete

STFT is decimated. the correspondence short-time sections $f_{nL}(m)$ are a sub set of $f_n(m)$ obtained by incrementing n by multiples of L.

2.2 Short-Time Fourier Synthesis: Overlap-Add(OLA) Method

The OLA method is motivated from the Fourier transform view of the STFT. The simplest method obtainable from the Fourier transform view is in fact not the OLA method. It is instead a method known as the inverse discrete Fourier transform(IDFT) method. In this method, for each fixed time, we take the inverse DFT of the correspondence frequency function and divide the result by the analysis window. This method is generally not favored in practical applications because the slightest perturbation in the STFT can result in a synthesized signal very different from the original. For example, consider the case where the STFT is multiplied by a linear phase factor the form $e^{j\omega n_0}$ with n_0 unknown. Then the IDFT for each fixed time results in a shifted version of the corresponding short-time section. Since the shift n_0 is unknown, dividing by the analysis window without taking the shift into account introduces a distortion in the resulting synthesized signal. In contrast, the OLA method, which we describe next, results in a shifted version of the original signal without distortion.

The OLA method is also best described in terms of the Fourier transform view. In the OLA method, we take the inverse DFT for each fixed time in the discrete STFT. However, instead of dividing out the analysis window from each of the resulting short-time sections, we perform an overlap-and-add operation between the short-time sections. This method works provided the analysis window is designed such that the overlap-and-add operation effectively eliminates the analysis window from the synthesized sequence. The OLA method is motivated by the following relation between a sequence and its discrete-time STFT:

$$x(n) = \left[\frac{1}{2\pi W(0)}\right] \int_{-\pi}^{\pi} \sum_{r=-\infty}^{\infty} X(r,\omega) e^{-j\omega n} d\omega \qquad (3.8)$$

where

$$W(0) = \sum_{n=-\infty}^{\infty} w(n) \qquad (3.9)$$

The OLA method carries out a discretized version of the operations suggested on the right of Eq. (3.8). That is, given a discrete STFT $X(n, k)$, the OLA method synthesizes a sequence $y(n)$ satisfying the following equation:

$$y(n) = \left[\frac{1}{W(0)}\right] \sum_{p=-\infty}^{\infty} \left[\frac{1}{N}\sum_{k=0}^{N-1} X(p, k)e^{j2\pi kn/N}\right] \qquad (3.10)$$

The term inside the rectangular brackets on the right is an inverse DFT that for each p gives

$$y_p(n) = x(n)w(p-n) \qquad (3.11)$$

The expression for y(n) therefore becomes

$$y(n) = \left[\frac{1}{W(0)}\right] \sum_{p=-\infty}^{\infty} x(n)w(p-n) \qquad (3.12)$$

which then reduces to

$$y(n) = x(n)\left[\frac{1}{W(0)}\right] \sum_{p=-\infty}^{\infty} w(p-n) \qquad (3.13)$$

In Eq.(3.12) we note that $y(n)$ will be equal to $x(n)$ provided

$$\sum_{p=-\infty}^{\infty} w(p-n) = W(0) \qquad (3.14)$$

Furthermore, if the discrete STFT has been decimated in time by a factor L, it can be similarly shown that if the analysis window satisfies

$$\sum_{p=-\infty}^{\infty} w(pL-n) = \frac{W(0)}{L} \qquad (3.15)$$

then x(n) can be synthesized using the following relation:

$$x(n) = \left[\frac{L}{W(0)}\right] \sum_{p=-\infty}^{\infty} \left[\frac{1}{N}\sum_{k=0}^{N-1} X(pL, k)e^{j2\pi kn/N}\right] \qquad (3.16)$$

Equation (3.15) is the general constraint imposed by the OLA method on the analysis window. It requires the sum of the analysis windows(obtained by sliding $w(n)$ with L-point increments at a time) to add up to a constant.

We can show that the OLA constraint in Eq.(3.15) is satisfied by all finite-bandwidth analysis window whose maximum frequency is less than $2\pi/L$, where L is the temporal decimation factor.

To see how finite-bandwidth analysis window satisfy the OLA constraint, suppose that the analysis window has maximum frequency ω_c, and consequently bandwidth $2\omega_c$. If we let $w'(p)$ denote the sequence $w(pL-n)$, then the OLA constraint in Eq.(3.15) can be rewritten as

$$W'(0) = \left[\frac{W(0)}{WL} \right] \qquad (3.17)$$

where $W'(\omega)$ denotes the Fourier transform of $w'(p)$. Noting that $w'(p)$ is a sampled version of $w(p-n)$, we can easily show that

$$W'(0) = \frac{1}{L} \sum_{k=-\infty}^{\infty} e^{-j(\omega - k2\pi/L)n} W(\omega - k2\pi/L) \qquad (3.18)$$

If there in no overlap between $W(\omega)$ and $W(\omega-k2\pi/L)$ at $\omega=0$, then Eq.(3.18) gives the OLA constraint expressed in Eq.(3.17). To have no overlap at $\omega=0$ between $W(\omega)$ and $W(\omega-k2\pi/L)$ it is easy to see that we must have $\omega_c < 2\pi/L$, where ω_c is the maximum frequency in $W(\omega)$. We conclude that any finite-bandwidth window whose maximum frequency is less than $2\pi/L$ will satisfy the OLA constraint in Eq.(3.15).

The transition width of main lobe of Hamming window is $8\pi/N$. N is window length. Therefore the decimation factor L should be less than $N/4$ when Hamming window is used as the analysis window.

2.3 Short-Time Fourier Transform Magnitude(STFTM) Analysis

In speech applications, the spectrogram that can be related to the magnitude of the STFT has played a major role. In particular, this representation is a non-negative time-frequency function. On the other hand, the STFT is generally a complex-valued function and for applications such as time scale modification of speech, estimation of the phase of this function is computationally difficult. In contrast, a number of techniques have been developed where the processed signal

is estimated from only the STFT magnitude(STFTM), thus circumventing the phase estimation problem.

The magnitude of the STFT is an alternative time-frequency signal representation. That the STFT is not a unique representation in all cases is easily seen from the simple observation that $x(n)$ and its negative, $-x(n)$, have the same STFTM. A one-sided sequence $x(n)$ can be recovered from its STFTM when the analysis window is nonzero over its finite duration and $x(n)$ satisfies the appropriate zero-gap restriction. The key to recovering $x(n)$ is the observation that $|X(n,\omega)|$ has additional information about the short-time sections of $x(n)$ besides their spectral magnitudes. This information is contained in the overlap of the analysis window positions. If the short-time section at time n is known, then the signal corresponding to the spectral magnitude of the adjacent section at time $n+1$ must be consistent in the region of overlap with the known short-time section. In particular, if the analysis window were nonzero and of length N_w, then after dividing out the analysis window, the first N_w-1 samples of the segment at time $n+1$ must equal the last N_w-1 sample of the segment from its first N_w-1 values, we could repeat this process to obtain the entire signal $x(n)$.

Like the STFT, the STFTM can be used for analyzing the time-varying spectral characteristic of a sequence. To carry out such STFTM analysis on a digital computer, we need to introduce the discrete STFTM. By sampling the frequency dimension of the STFTM, $|X(n, \omega)|$, we obtain the discrete STFTM, which is defined as $|X(n,k)|$, the magnitude of the discrete STFT.

2.4 Signal Estimation from Modified STFT or STFTM

In many applications it is desired to synthesize a signal from a time-frequency function formed by modifying an STFT or STFTM of a signal we wish to process. Such modifications may arise due to quantization errors in, for example, speech coding or purposeful time-varying filtering for signal processing application such as speech enhancement. An arbitrary function of time and frequency, however, does not necessarily represent the STFT or STFTM of a signal. This is because the definition of these transforms impose a structure on their time and frequency variations. In particular, because of the overlap between short-time sections,

adjacent short-time segments cannot have arbitrary variations. A necessary but not sufficient condition on these variations is that the short-time section corresponding to each time instant must lie within the duration of the corresponding analysis window. Even if this time-placement constraint is satisfied, a further condition that the STFT or STFTM must satisfy is that adjacent short-time sections should be consistent in their region of overlap. When the STFT or STFTM of a signal is modified, the resulting time-frequency function does not generally satisfy such constraints.

The synthesis method we explained in 2.2 was derived with the assumption that the time-frequency functions to which they are applied satisfy the constraints in the definitions of the STFT or STFTM. Given a function that does not satisfy those constraints, the synthesis method have no theoretical validity for their application. However, under certain conditions, those methods can be shown to yield reasonable results in the presence of modification.

2.4.1 Least-squares signal estimation from modified STFT

In this approach we estimate a signal whose STFT is closest in a least-square sense to the modified STFT. More specifically, we wish to minimize the mean-square error between the discrete-time STFT $X_e(n, \omega)$ of the signal estimate and the modified discrete-time STFT which we denote by $Y(n, \omega)$. This optimization results in the following solution for the estimated signal $x_e(n)$:

$$x_e(n) = \frac{\sum_{m=-\infty}^{\infty} w(m-n) f_m(n)}{\sum_{m=-\infty}^{\infty} w^2(m-n)} \qquad (3.19)$$

where $f_m(n)$ is the inverse Fourier transform of the frequency variation at time m of the modified STFT $Y(m, \omega)$. Since in practice we have only the discrete function $y(n,k)$, the short-time sections $f_n(m)$ can be obtained provided the frequency sampling factor N is large enough to avoide aliasing in the short-time sections. The specific distance measure used in the minimization is the squared error between $X_e(n, \omega)$ and $Y(n, \omega)$ integrated over all ω and summed over all n:

$$D\left[X_e(n,\omega), Y(n,\omega)\right] = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi}\int_{-\pi}^{\pi} |X_e(m,\omega) - Y(m,\omega)|^2 d\omega \qquad (3.20)$$

The solution in Eq.(3.19) extends in a simple manner to the case involving temporal decimation. Specifically, if L is the temporal decimation factor, then the solution in Eq.(3.19) becomes

$$x_e(n) = \frac{\sum_{m=-\infty}^{\infty} w(mL-n) f_{mL}(n)}{\sum_{m=-\infty}^{\infty} w^2(mL-n)} \qquad (3.21)$$

In general, the sum in the denominator of the right side of Eq.(3.21) is a function of n. However, there exist analysis windows $w(n)$ such that the sum in the denominator is independent of n. It should be noted that the sum in the denominator has the same form as the sum in the constraint equation (3.15) for the OLA method except that the analysis window is replaced by its square. That is, any window whose square satisfies the OLA constraint will make the denominator sum in Eq.(3.21) independent of n.

2.4.2 Least-squares signal estimation from modified STFTM

The least-square approach can also be used for signal estimation from the modified STFTM. The resulting method estimates a sequence $x_e(n)$ from a desired time-frequency function $|X_d(n, \omega)|$, which is a modified version of an original STFTM, $|X(n, \omega)|$. The method iteratively reduces the following distance measure between the STFTM $|X_e(n, \omega)|$ of the signal estimate and the modified STFTM $|X_d(n, \omega)|$:

$$D\left[X_e(n,\omega), Y(n,\omega)\right] = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi}\int_{-\pi}^{\pi} |X_e(m,\omega) - X_d(m,\omega)|^2 d\omega \qquad (3.22)$$

The solution is found iteratively because sa yet no closed-form solution has been discovered for $x_e(n)$ using the distance criterion in Eq.(3.22). The iteration takes place as follows. An arbitrary sequence(usually white noise) is selected as the first estimate $x_e^1(n)$ of $x_e(n)$. We then compute the STFT of $x_e^1(n)$ and modified it

by replacing its magnitude by the desired magnitude $|X_d(n, \omega)|$. From the resulting modified STFT, we can obtain a signal estimate using the method based on Eq.(3.19) in the previous section. This process continues iteratively, as shown in Fig. 3.1. In particular, the (i+1)st estimate $x_e^{i+1}(n)$ first obtained by computing the STFT $X_e^i(n, \omega)$ of $x_e^i(n)$ and replacing its magnitude by $|X_d(n, \omega)|$ to obtain $Y^i(n, \omega)$. The signal with the STFT closest to $Y^i(n, \omega)$ is found by using Eq.(3.19). All steps in the iteration can be summarized in the following update equation:

$$x_e^{i+1}(n) = \frac{\sum_{m=-\infty}^{\infty} w(m-n) \frac{1}{2\pi} \int_{-\pi}^{\pi} Y^i(m, \omega) e^{j\omega n} d\omega}{\sum_{m=-\infty}^{\infty} w^2(m-n)} \qquad (3.23)$$

where

$$Y^i(m, \omega) = |X_d(m,\omega)| \frac{X_e^i(m, \omega)}{|X_e^i(m, \omega)|} \qquad (3.24)$$

Although we restricted the preceding discussion to the discrete STFT, these results are easily extendable to the case where the STFT has been decimated in time.

3. A New Pitch Frequency Modification Algorithm

A block diagram of the proposed algorithm is shown in Fig.3.1.

3.1 Spectrum Envelope Extraction

Homomorphic deconvolution is first applied to get the source component $G(mL, \omega)$. The unique concept in this block is the lifter (referred to as "comb lifter" in this paper) that passes all cepstrum except cepstrum in the pitch frequency region. This is based on the idea that the parameters which contribute only to the pitch frequency should be changed, while the rest of the parameters should remain unchanged to maintain high quality.

Fig.3.3(a), for example, shows a magnitude spectrum of spectrum envelope and source that were separated by a conventional cepstrum lifter which passes all low-quefrency part cepstrum($n<30$). Also, Fig.3.3(b) shows them separated by the "comb lifter" which passes all cepstrum except those 20 points (1.67msec in quefrency, at 12KHz sampling) to each side of the pitch period. Although the magnitude spectrum of the source component in Fig.3.3(a) has a flat spectrum envelope, the shape of the spectrum is not uniform; i.e., at some points (a or b in Fig.3.3(a)) the spectrum dips. On the other hand, the magnitude spectrum of the source component in Fig.3.3(b) is uniform and similar to a sine curve. This indicates that pitch modification to the output of the "comb lifter" makes it possible to reconstruct a signal without any side-effect in the high frequency region. Judging from the preliminary listening test, the "comb lifter" makes modified speech clear.

3.2 Pitch Frequency Modification

To change the pitch frequency a linear interpolation is performed by introducing a modification-factor k to both the real and imaginary parts of the source component $G(mL, \omega)$. The factor k is defined as the ratio of the original pitch frequency to the desired pitch frequency. To achieve high spectrum resonance, a 512-point FFT is used for 256-point speech data. Therefore, an additional 256 zeros are set in the data array for FFT as shown in Fig.3.4(a). When the pitch frequency is raised, the resultant data that exceed the maximum frequency-band are discarded and when it is lowered, unknown data near the maximum frequency are regenerated by mirror image copying of the lower part of the spectrum. We get modified source spectrum $G_m(mL, \omega)$ through the above procedures.

The copying in the pitch-lowering modification has the side-effect shown in Fig.3.4(b); i.e., the part where zeros are set in the original speech has non-zero values after the modification. To compensate for this side-effect, modified spectrum $G'(mL, \omega)$ is obtained by the following equation.

$$G'(mL,n)=G'_m(mL,n)\ \frac{|G_m(mL,\omega)|}{|G'_m(mL,\omega)|} \qquad (3.25)$$

where $G'_m(mL, \omega)$ is the spectrum of the residual signal, shown in Fig.3.4(b), whose non-zero part is replaced by zero. This procedure, as shown in Fig.3.4(c), brings the non-zero value close to zero.

3.3 Phase Adjustment

After multiplying the modified source component $G'(mL, \omega)$ by the spectrum envelope component $|V(mL, \omega)|$, modified speech is obtained frame by frame by the inverse Fourier Transform. Fig.3.5 shows windowed speech signals for successive frames. The left speech signals are original speech and the right ones are pitch-raised speech. These signals are overlap added using the Eq.(3.21) in section 2 to make a modified speech signal $x'(n)$. It is apparent, as shown in Fig.3.5, that the phase of pitch-raised speech is not continuous between frames because of the linear interpolation in the previous block. To cope with this, a method of variable window shift is used; i.e. when speech is analyzed with window shift L and modified with a modification-factor k, the window shift is replaced by $L'=L/k$ instead of L in overlap adding.

3.4 Duration Adjustment

The speech signal modified by the previous block is generally different in duration from the original because of the window shift control. One purpose of this block is to compensate for this side-effect, the other is to modify speech duration. Duration modification is performed by the signal reconstruction algorithm from the modified STFT magnitude proposed by Griffin and Lim[Griffin, 1984] which was described in section 2.4.2. White noise is used as the initial value $x(n)$. The compensation procedure is as follows. First the modified speech $x'(n)$ is again analyzed with window shift L', then reconstructed with window shift L.

4. Analysis-Synthesis Experiment

To confirm potential performance of the proposed algorithm, a speech analysis-synthesis experiment was carried out. Figure 3.6 shows spectrograms and speech waves of an original speech, analysis-synthesis speech by the proposed method, by cepstrum vocoder[Imai, 1980], by LPC vocoder. The speech was uttered a male speaker. The experiment conditions are shown in Table 3.1.

In Fig. 3.6, it is observed that the spectrogram of both cepstrum and LPC vocoders is too simplified, but that the spectrogram of the proposed method is very close to that of the original. In terms of the speech waveform, cepstrum and LPC vocoders synthesize speech which looks like impulse. The proposed method well reconstructed pitch structure which are different from both the original speech and the outputs vocoders. Judging from the informal listening test, we could not find out difference between original speech and the analysis-synthesis speech by the proposed method.

5. Speech Modification Experiment

In the pitch frequency modification, the proposed algorithm also needs time modification. In this section, time modification experiment is first carried out, then pitch modification experiment is carried out to evaluate the overall performance.

5.1 Time Modification

Speech uttered by a male speaker was expanded by 1.3 and compressed by 0.7. The experiment condition is shown table 3.1. Figure 3.7 shows estimation error according to a iteration number.The estimation error monotonously decrease according to iteration, especially it quickly decreases during the first five iterations. A small amount of reverberation was detectable in the estimated signal, but the signal was still high quality. The estimation error of compression factor 0.7 converges to higher value than the error of expansion factor 1.3.

However, there were no difference between the two in its quality.In terms of iteration number, there is no improvement in its quality after six times.

5.2 Pitch Frequency Modification

In this section the effect of the proposed algorithm on pitch modification is discussed. Two pitch frequency modifications are evaluated by listening tests. One modification is uniform raising or lowering, and the other is non-uniform modification.

5.2.1 Experiment method

The performance of the proposed algorithm was compared with the cepstrum vocoder, because its quality was the best among the vocoder speech tried. Table 3.1 shows the analysis-synthesis conditions. Sets of two kinds of stimuli are randomly presented to 8 listeners who are not familiar with the synthesized speech. One pair consists of original speech and speech modified by the proposed algorithm, the other consists of original speech and speech modified by the cepstrum vocoder. The listeners are asked to score the pairs according to the similarity between the original and modified speech. The modifications for producing evaluation test speech samples are as follows.

(1) Uniform modification: Pitch frequency of seven words was modified uniformly by modification-factors 0.9 and 1.1.

(2) Non-uniform modification: Like Chinese which is a tonal language, some Japanese words with the same phonetic structure differ in meaning according to pitch contour. Four pairs of words that have this property are used in non-uniform modification. Fig.3.6 shows, for example, the original pitch frequency, the modified pitch frequency and the modification-factor for each frame. The words are /ueru/, /shouhiN/, /seNryou/, and /deNki/.

5.2.2 Experiment results

Table 3.2 shows the experiment results. The score that listeners gave to the speech of the proposed algorithm is divided by the score obtained from the vocoder speech. This ratio is given in the table. The reason pitch lowering is less satisfaction than pitch raising is the regeneration by copying described in Section 3.2. The table indicates that in all modifications the proposed algorithm can modify speech better than the vocoder algorithm.

6. Conclusion

We proposed a new speech modification algorithm. The advantages of this algorithm are listed below;

(1) This algorithm needs no phase unwrapping which is the most complex and critical procedure in the conventional method.
(2) This algorithm is easy to implement in an automatic system because explicit pitch frequency extraction is not required.
(3) The quality of synthesized speech is very high and natural because residual signals are used as excitation.
(4) This algorithm makes it possible to modify the spectrum envelope in a non-parametric way because it is represented by FFT magnitude.

The listening test reveals that the proposed algorithm can reproduce high quality speech sounds in pitch frequency modification for both uniform and non-uniform pitch modification.

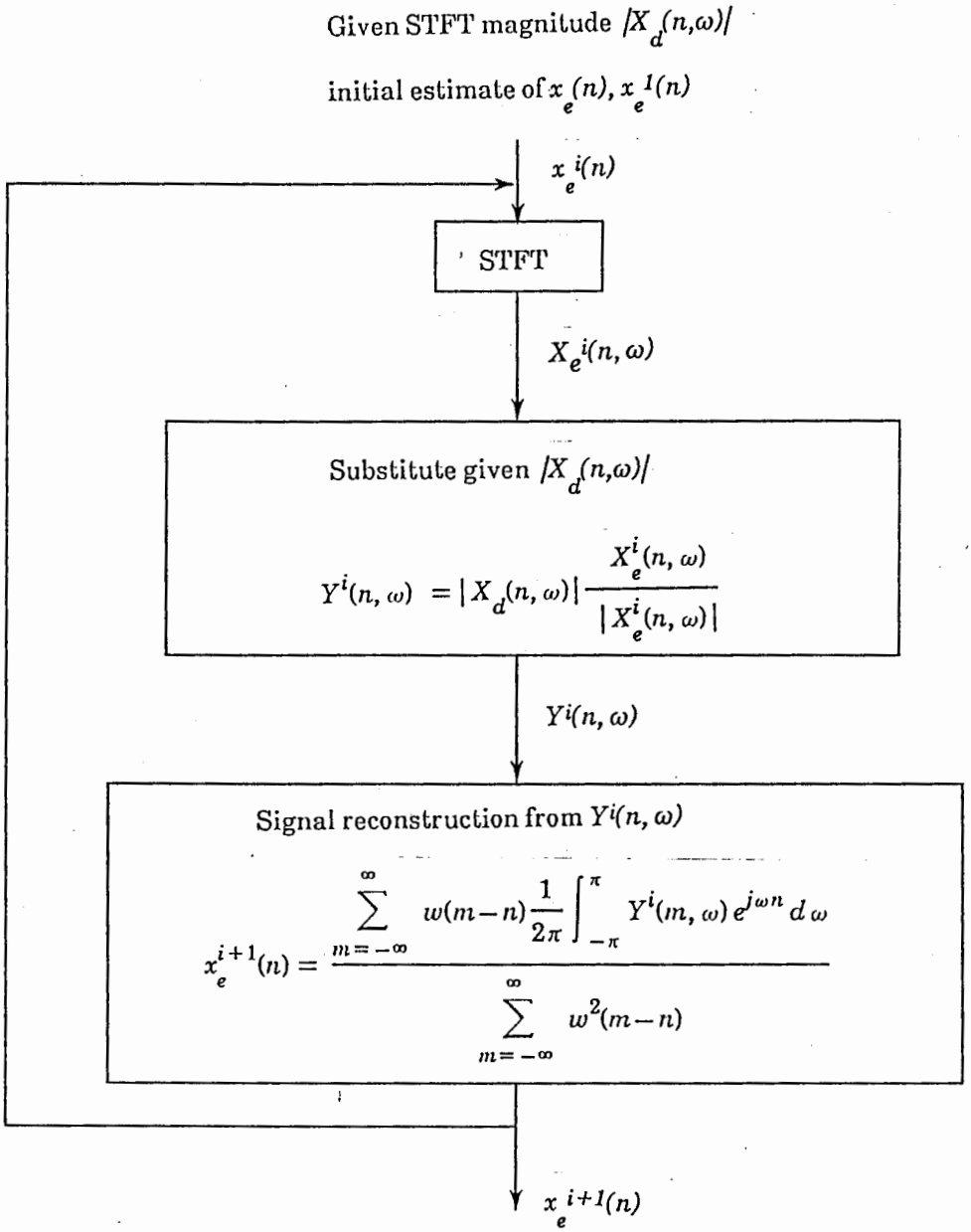


Fig.3.1 Signal reconstruction algorithm from modified STFT magnitude

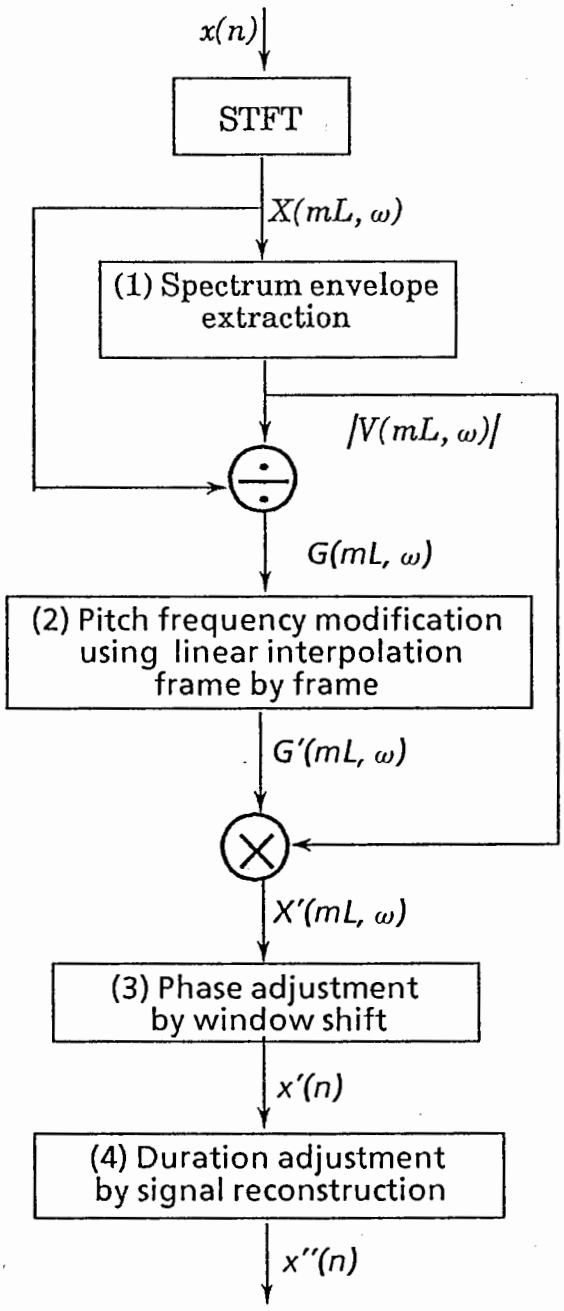
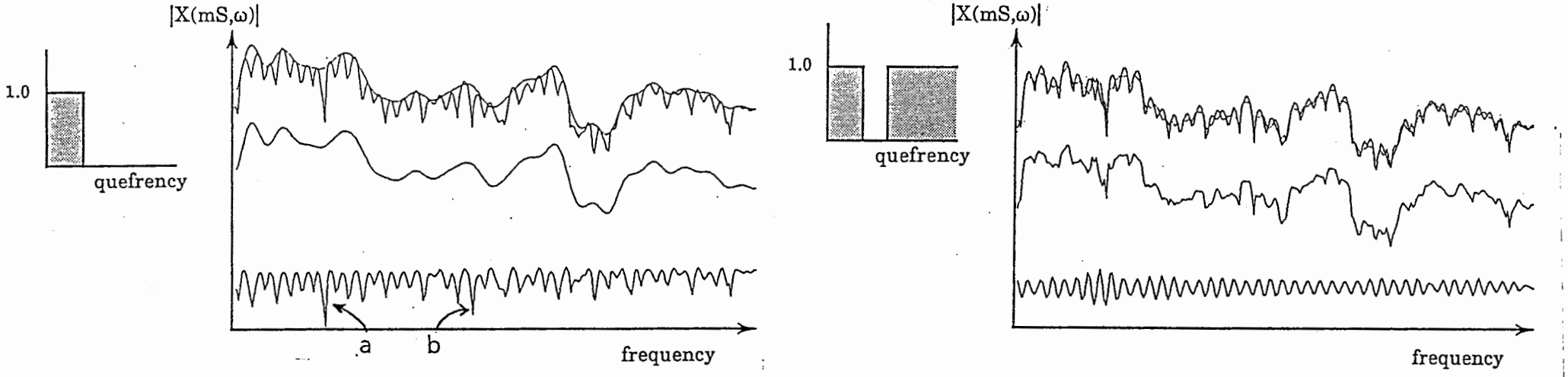


Fig.3.2 Diagram of speech modification method



(a) Spectrum envelope by cepstrum lifter

(b) Spectrum envelope by "Comb" lifter

Fig.3.3 Cepstrum lifters

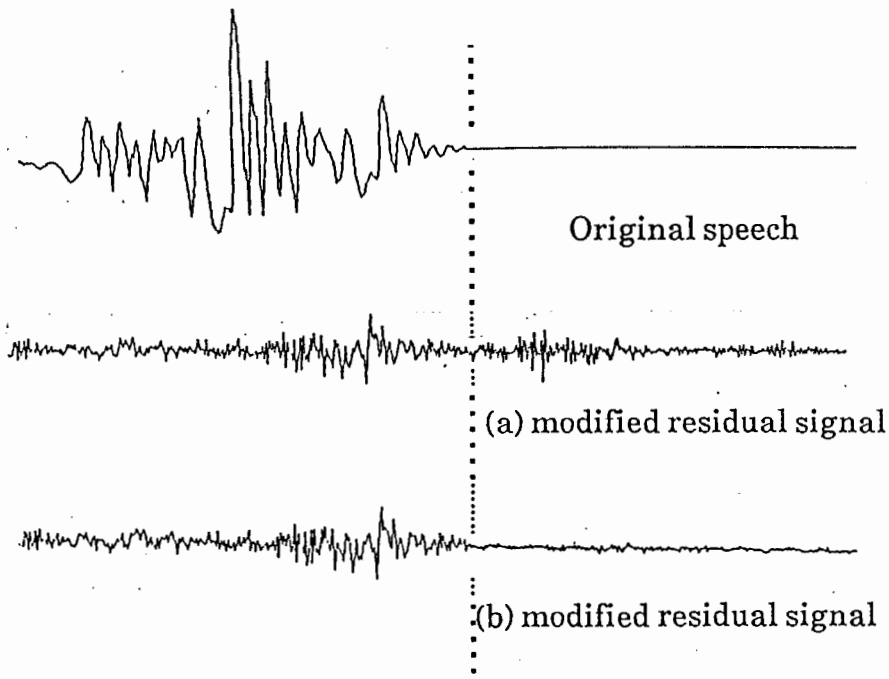


Fig.3.4 Original speech and modified residual signal

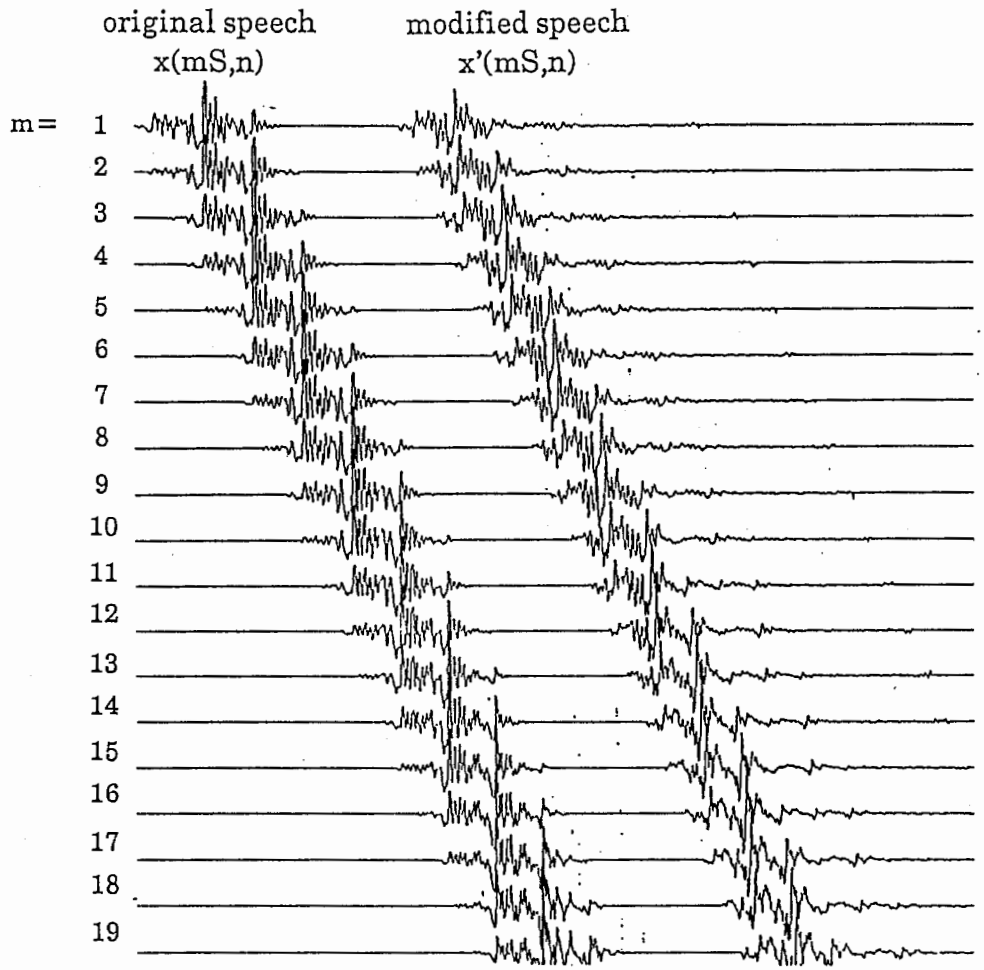


Fig.3.5 Windowed original speech and windowed modified speech

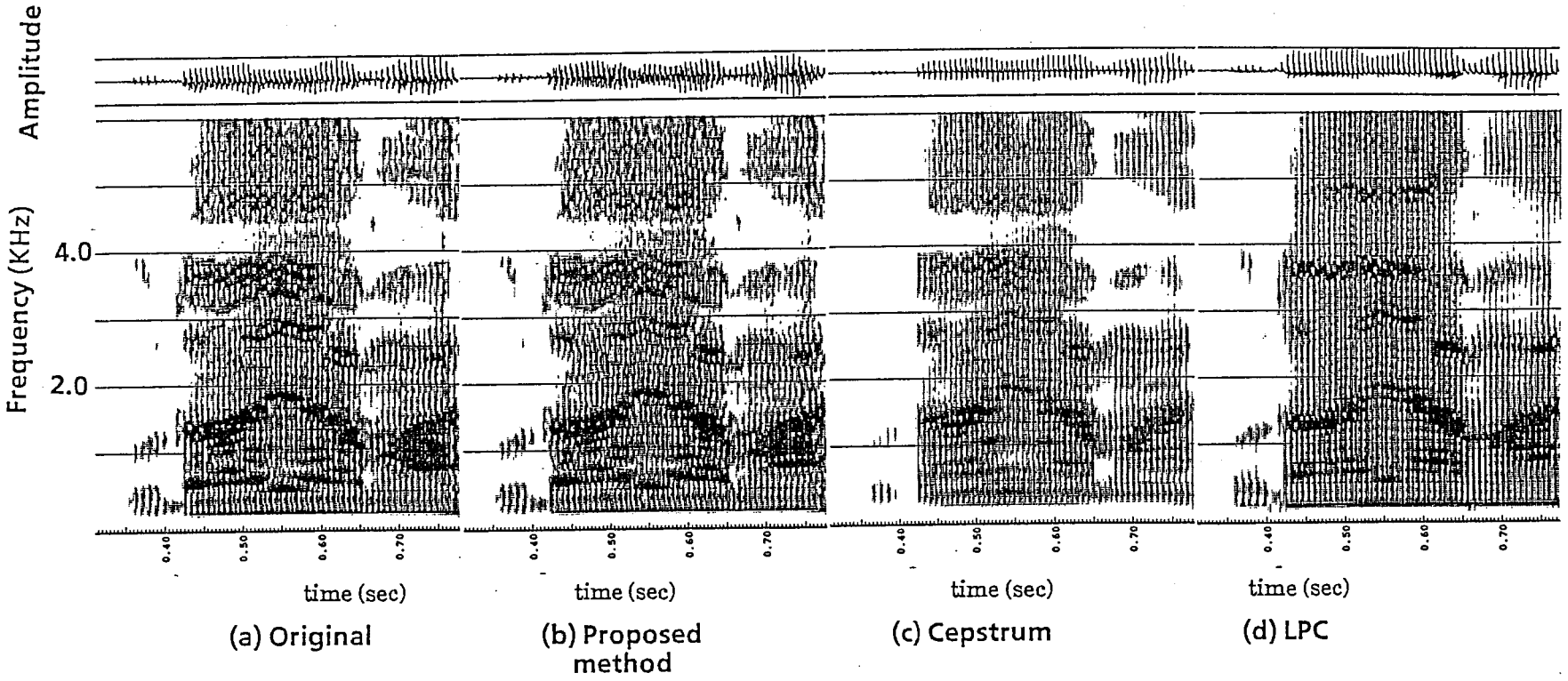


Fig.3.6 Spectrograms of synthesized speech

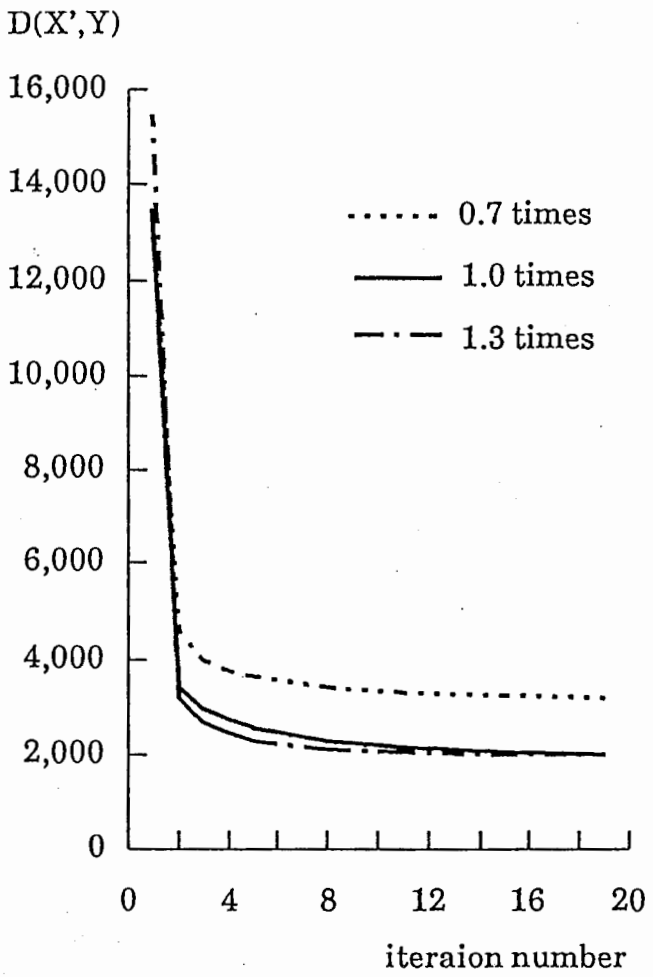


Fig.3.7 Estimation error versus iteration number

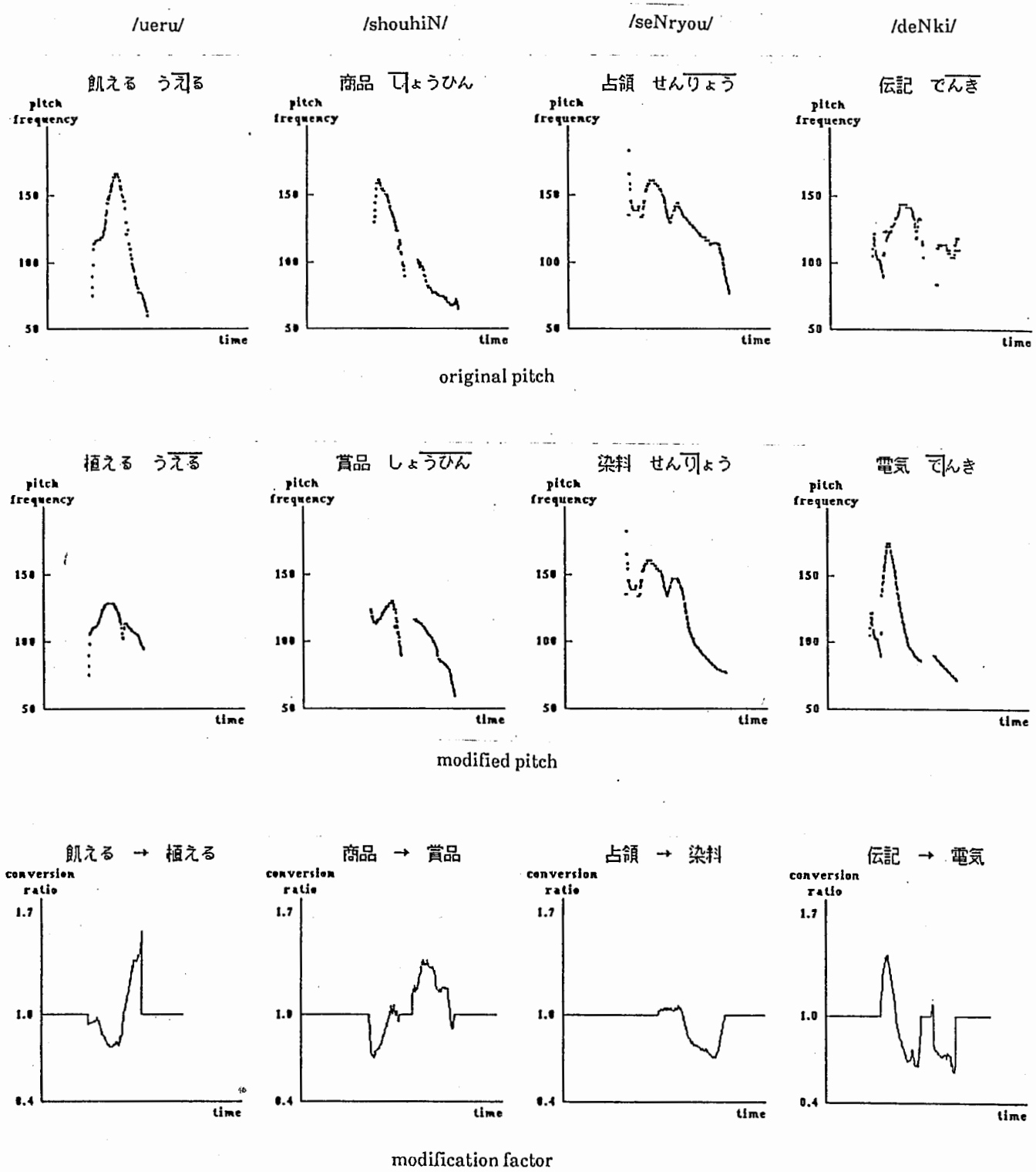


Fig.3.8 Pitch-modification-factor for non-uniform modification

Table 3.1-0 ExperimentConditions(common)

A/Ddata	12KHz sampling, 16bit
window	Hamming window
window length	256points (21.3msec)

Table 3.1-1 Experiment Conditions (proposed method)

window shift	32points (2.7msec)
FFT points	512points
iteration for reconstruction	20

Table 3.1-2 Experiment Conditions (cepstrum synthesis)

window shift	60points (5.0msec)
cepstrum order	30
synthesis filter	LMA

Table 3.1-3 Experiment Conditions (LPC synthesis)

window shift	60points (5.0msec)
cepstrum order	14

Table 3.2 Experiment result

	ratio
uniform (0.9 times)	1.43
uniform (1.1times)	1.68
non-uniform	1.41

Chapter 4

Cross-Language Voice Conversion

1. Introduction

In recent years, there are many opportunities to communicate with speakers of other languages. We must make an effort to communicate in other languages, sometimes making mistakes in understanding or even, if we don't know the language, not understanding anything at all. A system using the latest information processing technology to overcome this language barrier would be very useful. One such system is an automatic telephone interpretation system: i.e., a facility that enables a person speaking in one language to communicate readily by telephone with someone speaking another language [Kurematsu, 1987]. Our laboratory, ATR Interpreting Telephony Res. Labs., is dedicated to research aimed at developing an interpreting telephony system.

This system consists of three constituent technologies: speech recognition, machine translation, and speech synthesis. An interpreting telephone will recognize Japanese speech, translate it into English, then synthesize English speech. It will, of course, also work in the opposite direction. To develop an interpreting telephone, there are many issues to be solved in the constituent subsystems. In this chapter, we will discuss speech individuality control in an interpreting telephone. Because we can recognize a speaker over a conventional telephone line, we would like also to retain speaker individuality in an interpreting telephone. In other words, the quality of the synthesized English speech should be changed to sound as if the Japanese speaker uttered the English. We call the problem "cross-language voice conversion" [Abe, 1990a][Abe, 1990b], because speech individuality would be preserved across different languages.

In section 2, spectral differences between English and Japanese are investigated using speech from a bilingual speaker. In section 3, we propose cross-language voice conversion algorithms and evaluate their performances.

2. Japanese Spectrum Space vs. English Spectrum Space

To examine spectrum differences in different languages, speech uttered by a bilingual speaker was analyzed. The material makes it possible not only to eliminate spectrum differences caused by different speakers but also to specifically focus on the spectrum differences caused by the two languages. The results of this section will be used to develop a cross-language voice conversion based on codebook mapping in section 3. The following points are discussed.

① How much does the spectrum space increase to deal with more than one language?

Codebook size is important in our voice conversion algorithm because spectrum characteristics of a speaker are represented by code vectors of the speaker's codebook. Codebook size for mixed speech of English and Japanese is examined in section 2.2.

② Are there any spectra which characterize certain English or Japanese sounds?

Voice conversion is performed by replacing a speaker's code vector with the corresponding code vector of another speaker. The requirement that every English code vector has a corresponding Japanese code vector is examined in section 2.3.

③ Which phonemes contain the spectra?

In section 2.4, we examine phonemes that contain code vectors which predominantly occur in English or Japanese.

④ How important are the spectra from a perceptual point of view?

In section 2.5, the spectral differences are investigated from a perceptual point of view.

2.1 Speech Data and Analysis Method

To investigate the spectrum difference between Japanese and English, speech uttered by a bilingual speaker was collected. The bilingual speaker whose mother and father are Japanese and German, respectively, was born and brought up in Japan. To select the bilingual speaker, we especially paid attention to his pronunciation. Native speakers have judged his English and Japanese pronunciation to be of native speaker level. The speaker read a list of 216 phonetically balanced Japanese words and 328 phonetically balanced English words. Three male and three female Japanese speakers also read the list of 216 phonetically balanced Japanese words.

Codebooks were generated using the Linde-Buzo-Gray(LBG) algorithm[Linde, 1980]. Table 4.1 shows analysis parameters.

2.2 How Much does the Spectrum Space Increase to Deal with More than One Language?

2.2.1 Experimental method

Codebook sizes were examined using a spectrum distortion measure, WLR(Weighted Likelihood Ratio), is defined by the following equation [Sugiyama, 1981]:

$$D = \sum_{i=1}^{n} (r_i - r_i')(c_i - c_i') \qquad (4.1)$$

Here, r_i and r_i' are the i-th autocorrelation coefficients and c_i and c_i' are the i-th LPC cepstrum coefficients. This measure enhances the contrast between peaks and valleys of the LPC spectral envelope.

Six codebooks were generated for the following data sets.

(1) English and Japanese words uttered by the bilingual speaker

(2) English words uttered by the bilingual speaker
(3) Japanese words uttered by the bilingual speaker
(4) Japanese words uttered by one male and one female speaker
(5) Japanese words uttered by two male speakers
(6) Japanese words uttered by one male speaker

2.2.2 Experimental results

Fig.4.1 shows spectrum distortions for data from (1) to (6) according to codebook size. Because the spectrum distortions in (2), (3) and (6) are almost the same, we can use the same codebook size in both English and Japanese to represent spectrum characteristics of a speaker. The distortion of (1) in an 8-bit codebook almost equals the distortion of (2), (3) and (6) in 7-bit codebooks, and is smaller than the distortion in (4) and (5). This indicates that, when a codebook is generated for speech from English and Japanese, its codebook size should be almost twice as large as the codebook size of English or Japanese, but does not have to be as much as the codebook size of two speakers.

2.3 Are There any Spectra which Characterize Certain English or Japanese Sounds?

2.3.1 Experimental method

To investigate if there are any code vectors(spectra) which characterize certain English or Japanese sounds, and to know how much code vectors overlap in different languages, an experiment was carried out. Experimental procedures are as follows:

(1) A codebook was generated using the mixed speech from the following category pairs:

(A)English vs. Japanese
(B)male speaker vs. female speaker (in Japanese)
(C)male speaker 1 vs. male speaker 2 (in Japanese)
(D)word set 1 vs. word set 2 (uttered by the same speaker in Japanese)

(2) All data from the each category pair was vector quantized using the codebook.

(3) We count how many times a code vector of the codebook occurred in two categories.

The distribution distance between the above categories is calculated using Kullback's divergence[Kullback, 1970] defined as follows:

$$D = \sum_{i=1}^{r} [P(a_i|\omega_1) - P(a_i|\omega_2)] \log \frac{P(a_i|\omega_1)}{P(a_i|\omega_2)} \qquad (4.2)$$

where, ω_1 is category 1, ω_2 is category 2, r is a codebook size, and $P(a_i|\omega_j)$ is the posteriori probability of the code vector a_i in category ω_j.

2.3.2 Experimental results

Table 4.2 shows Kullback's divergence for each category pair. Kullback's divergence indicates the overall distance between two distributions; the larger the value, the more the two categories are separated. Therefore, the data uttered by the male speaker and the female speaker are well separated and the data uttered by the same speaker is difficult to separate. Judging from the value of the English-Japanese pair, the two categories show more overlap than separation.

To show this visually, scatter plots are shown in Figs. 4.2, 4.3, 4.4, 4.5. Points in the figures show code vectors, and are plotted according to the occurrence number in each category. Fig.4.2 indicates that some code vectors have a tendency to occur more frequently in Japanese, and, conversely, other in English.

2.4 Which Phonemes Contain the Spectra which Characterize a Language?

2.4.1 Experimental method

The results in 2.3 indicate that some code vectors(spectra) predominantly occur in either English or Japanese. In this section, we examined correspondences between phonemes and code vectors.

The phonetic transcription was aligned for English and Japanese utterances produced by the bilingual speaker. For English, this was done by CASPAR, an automatic alignment system developed at MIT[Leung, 1985] and errors were corrected by hand. For Japanese, alignment was done by hand according to ATR's labeling style[Kuwabara, 1989].

2.4.2 Experimental results

Code vectors which predominantly occurred in English or in Japanese are summarized in Table 4.3 in terms of the percentage of constituent phonemes for each code vector. Figure 4.6~4.13 shows the LPC spectrum envelope of some representative code vectors and percentage of constituent phonemes of each code vector. The following are the characteristics of each code vector(A-H). Here, phonetic symbols for English and Japanese are used in accordance with TIMIT conventions[Garofolo, 1988] and Roman letters respectively. As a reference, the F1-F2 relationship for vowels in English and Japanese is shown in Fig.4.14[Zue, 1985][Umeda, 1957].

(A) Vowels /ɔ/,/a/,/ə/. F1 and F2 are very close. This formant structure is rarely found in code vectors which occurred frequently in Japanese.

(B) Vowel /æ/. This is a typical formant structure of /æ/. It is said that Japanese has no vowel of this formant structure.

(C) Consonants /č/,/ǰ/,/š/. These are voiceless consonants but the formant structure is very clear in the spectrum envelope.

(D) Consonants /f/,/t/. These are voiceless consonants but the formant structure is very clear.

(E) Liquid /r/. F2 and F3 are very close. This is a typical /r/ in English.

(F) Vowel /i/. This is a typical formant structure of /i/. F1 and F2 are very far. English /i/ is typically more centralized.

(G) Vowel /u/. This is a typical formant structure of /u/. F1 is relatively low and F2 is midfrequency. English /u/ is typically more centralized.

(H) Nasal /N/. The spectrum envelope has two peaks in low frequency and high frequency. In the English nasal, there is no such high frequency peak.

2.5 How Important are the Spectra which Characterize a Language from the Perceptual Point of View?

2.5.1 Experimental method

The experimental results in 2.3 and 2.4 showed that there are spectral differences between Japanese and English. These differences were examined using a perceptual experiment.

First, two codebooks were generated for the bilingual speaker, one from English (English codebook) and the other from Japanese(Japanese codebook). Two kinds of speech were synthesized using the bilingual speaker's English speech, one is coded by the English codebook, then decoded(CEDE), the other is coded by the Japanese codebook, then decoded(CJDJ). Because the CJDJ is represented by code vectors of Japanese codebook, we can predict that CJDJ will not sound like English if the spectral differences of English and Japanese code vectors are perceptually large enough. CEDE and CJDJ pairs were synthesized by a LPC vocoder for twenty-eight words which contain all English phonemes at least once. All word pairs were presented to 8 native American listeners (4 males, 4 females) over headphones. The listeners were asked to judge whether there was a difference between the pairs. If not, the next word was presented. If a difference was noticed, listeners indicated which sounded more like English and gave a reason for their choice.

2.5.2 Experimental results

Table 4.4 shows how often the distinction between the two words is judged correctly or incorrectly or judged to be indistinguishable. Since the CEDE should sound more like English than the CJDJ words because CEDE words were coded by English codebook, we use the term "correct" when CEDE is judged to be better than CJDJ.

The results tended to be word dependent. Some words had a tendency to be judged indistinguishable, while others were judged as either correct or incorrect more than half the time as shown in the table. In Table 4.5, the words used in the experiment are classified into these three categories. That half the words are

judged to be indistinguishable is not unreasonable, because, as shown in 2.3, English and Japanese code vectors almost overlap.

Judging from the results in section 2.4, phonemes listed in Table 4.3 are expected to sound worse. However, such a tendency can not be shown in Table 4.5. The reasons are: (1) Because the vectors were matched to the input English frame by frame, the sequence in CJDJ preserved the dynamic characteristics of English such as formant frequency trajectories. (2) Phonemes listed in Table 4.3 are not composed exclusively of code vectors which predominantly occur in English, but also include vectors commonly occurring in Japanese.

2.6 Summary

Speech uttered by a bilingual speaker was analyzed. Experimental results are as follows:

(1) Codebook size for mixed speech from English and Japanese is almost twice as large as the codebook size of either English or Japanese, but does not have to be as much as the codebook size of two speakers.
(2) Although many code vectors occurred in both English and Japanese, some code vectors have a tendency to predominantly occur in Japanese or in English.
(3) Code vectors which predominantly occurred in English are contained in /r/,/æ/,/f/,/š/, and code vectors which predominantly occurred in Japanese are contained in /i/,/u/,/N/.
(4) Judging from listening tests, English speech decoded by Japanese codebook can be also recognized as English.

When we apply the voice conversion algorithm based on codebook mapping to the cross-language voice conversion, we have to pay attention the fact there are some code vectors which predominantly occurred in English or Japanese. One solution is to teach Japanese or English speakers how to pronounce the phonemes which contain such code vectors. The other approach is to neglect such code

vectors, because the result in 2.5 indicates that they are not perceptually important.

3.Cross-Language Voice Conversion Experiment

In this section, we propose cross-language voice conversion algorithms based on codebook mapping, and discuss its performance.

In the final stage of an interpreting telephony system, English is synthesized by a synthesis-by-rule system using output of the translation system. Fig.4.15 shows the block diagram of our cross-language voice conversion model. MITalk, a synthesis-by-rule system for English[Allen, 1979], is presently used because it is easily obtainable. The aim is to modify MITalk's speech so that the output carries the voice characteristics of a given Japanese speaker's speech. Strategies we take here are (1)preserve dynamic characteristics of MITalk's speech, because the results in section 2.5 imply that dynamic characteristics of English help synthesized speech sound like English, (2)change MITalk's speech spectrum into the Japanese speaker's speech spectrum, because the static characteristics of the speech spectrum contain important information of speaker individuality.

In terms of the strategy (2), we apply the voice conversion algorithm proposed in chapter 2. Because a mapping codebook is generated by supervised training, it is impossible to directly generate a mapping codebook using speech uttered by a Japanese speaker and speech synthesized by MITalk system. To solve the problem is a main topic of this section.

3.1 Methods to Make a Mapping Codebook across Different Languages

A policy to design a cross-language voice conversion method is not to ask a speaker to pronounce a non-mother language. In other words, a method is

designed to be useful for anyone who can not speak the non-mother language at all.

3.1.1 Method1: Synthesize Japanese by MITalk

To generate a mapping codebook, Japanese words are used as training data, i.e., Japanese speakers utter Japanese words and MITalk system synthesizes the same Japanese words. Two kinds of speech are synthesized. One, MITalk-E, is synthesized using input strings selected so that the output sounded as much like the Japanese word as possible, but using the default MITalk rules for English. The other, MITalk-Ed, is synthesized using duration control rules for Japanese. This modification is performed to well find out code vector correspondence in DTW, because durations are very different in English and Japanese.

3.1.2 Method2: Generate a mapping codebook through a bilingual speaker

To generate a mapping codebook, we propose making use of a bilingual speaker's speech as a bridge. First, four codebooks are generated, i.e.; using English uttered by an English speaker, using Japanese uttered by a Japanese speaker, and using English and Japanese uttered by a bilingual. Then, the following mapping codebooks are generated: one is between an English speaker and a bilingual speaker using English utterances(English mapping codebook), the other is between a Japanese speaker and the bilingual speaker using Japanese utterances(Japanese mapping codebook). Using the above codebooks, cross-language voice conversion is performed as follows (ref. Fig.4.16)

(1) Speech uttered by the English speaker is vector quantized using his codebook.
(2) According to the code vector obtained in (1), a code vector is selected from the English mapping codebook.
(3) The English code vector is approximated by fuzzy VQ using the bilingual speaker's Japanese code vectors. Fuzzy VQ[Ruspini, 1970], which is one technique to approximate a vector X by linear combination of code vectors V_i, is defined using the following equation.

$$u_i = 1 / \left[\sum_{j=1}^{k} \left(d_i / d_j \right)^{1/(m-1)} \right] \tag{4.3}$$

$$X' = \sum_{i=1}^{k} \left[\left(u_i \right)^m \cdot V_i \right] / \sum_{i=1}^{k} \left(u_i \right)^m \tag{4.4}$$

where u_i; fuzzy membership function. $u_i \in [0,1] \ \forall i$

$d_i = \| X - V_i \|$. V_i is a codeword in codebook V.

m: fuzziness. ($m=1.6$)

X': decoded spectrum of input spectrum X

k: nearest neighbors ($k=6$)

(4) Hypothesizing that the fuzzy membership functions obtained in step (3) are preserved in a target Japanese speaker's spectrum space, the spectrum is generated as a linear combination of code vectors in the Japanese mapping codebook.

3.2 Performance Evaluation

3.2.1 Performance evaluation of Method1

3.2.1.1 Evaluation method for Method1

In Method1, consistency of code vector correspondences is important to get good performance. To measure the consistency, mutual information is calculated by considering the voice conversion a transmission through an information channel(Fig.4.17).

The input alphabet A = { a_i } , i=1,2,...,r, consists of the code vectors of speaker A, and the output alphabet B = { b_j } , j=1,2,...,r, consists of the code vectors of speaker B. Mutual information I(A;B) is defined by the following equations;

$$I(A;B) = H(A) - H(A|B) \tag{4.5}$$

where,

$$H(A) = \sum_{i=1}^{r} P(a_i) \log \frac{1}{P(a_i)} \tag{4.6}$$

$$H(A|B) = \sum_{i=1}^{r} P(b_i) \sum_{j=1}^{r} P(a_i|b_j) \log \frac{1}{P(a_i|b_j)} \tag{4.7}$$

$P(a_i)$: a probability of code vector a_i of speaker A
$P(a_i|b_j)$: a posteriori probability of the input symbol a_i

In this formulation, the larger the value, the more consistent correspondences there are between the two code vectors.

3.2.1.2 Performance of Method1

The mapping codebooks are generated for all combinations of all speakers, i.e., male Japanese speaker, female Japanese speaker, MITalk-E, and MITalk-Ed. Fig.4.18 shows the mutual information for each speaker pair. The results indicate that the voice conversion is best in the J-J pair, and is poorest performance in the E-J pair. The reasons are as follows; (1)Because MITalk-Ed is given Japanese phoneme duration, the correspondence between the J-Ed pair is more consistent than that of the E-J pair. (2)Because MITalk-E and MITalk-Ed have the same rule of spectrum pattern generation, the distortion measure is more reliable in the E-Ed pair than in the E-J. Judging from these results, an adjustment of duration control to Japanese is necessary to improve voice conversion performance, but not enough. Furthermore, the distortion measure should be carefully selected when distortion is calculated between human and synthesizer.

Judging from the informal listening test, the cross-language voice conversion performance is judged to be worse than voice conversion from Japanese. The degradation is caused by the inconsistency of code vector correspondences.

3.2.2 Performance evaluation of Method2

3.2.2.1 Evaluation of fuzzy VQ approximation

Because fuzzy VQ approximation is newly introduced in Method2, we discuss the effect of fuzzy VQ approximation. To evaluate the effects, we used bilingual speaker's speech, because English which is obtained by fuzzy VQ approximation using Japanese can be compared with English which uttered the same speaker. Four codebooks are generated for two bilingual speakers, i.e; male-English, male-Japanese, female-English, female-Japanese. Table 4.1 shows the experiment conditions.

Table 4.6 shows the distortion when English uttered by a bilingual speaker is vector quantized or fuzzy vector quantized using the Japanese codebook or the English codebook. Fig 4.19 shows the distortion for every phoneme when English is vector quantized or fuzzy vector quantized using the Japanese codebook. Judging from these results, fuzzy VQ is useful in approximating the English spectrum using the Japanese codebook, and the approximation is effective for all phonemes.

To confirm the hypothesis that the fuzzy membership function is preserved in the target speaker's spectrum space, an experiment is performed as follows:

(1)A mapping codebooks are generated for male-Japanese, female-Japanese pairs.
(2)English uttered by the male speaker is vector quantized or fuzzy vector quantized using his Japanese codebook.
(3)To generate a converted spectrum, two methods are applied: using a mapping codebook, and using a mapping codebook and the fuzzy membership function with the hypothesis.
(4)Spectrum distortion is calculated between the target female speaker's English speech and converted speech.

If the fuzzy membership function was preserved in the target speaker's spectrum space, the spectrum distortion using fuzzy VQ would decrease.

All combinations of two bilingual speakers and two languages are investigated using the above procedures. The experiment results are shown in Table 4.7. The results indicate that the hypothesis is valid.

3.2.2.2 Performance of Method2

Codebooks, nine in total, are generated for every person and language, i.e; four for Japanese speakers(2 male and 2 female), four for bilingual speakers(male-English, male-Japanese, female-English, female-Japanese), and one for MITalk. Then mapping codebooks are generated for all speaker pairs. Table 4.1 show the experiment conditions.

Table 4.8 shows the experiment results. In terms of codebook generation, spectrum distortion in MITalk is very much less than in a human, because the variety of spectrum patterns is quite restricted and the speech is synthesized by formant synthesizer. The distortion of the mapping codebook between a human and MITalk is considerably larger than the mapping between human.

Judging from the informal listening test, cross-language voice conversion performance is judged to be worse than voice conversion from Japanese. The reasons are (1)mapping codebooks are used twice, (2)speech synthesized by synthesis-by-rule system is used instead of human speech. In terms of the performance, there is no difference between the Method1 and the Method2.

4. Conclusion

To apply a voice conversion algorithm based on codebook mapping to the cross-language voice conversion, speech uttered by a bilingual speaker was analyzed. Experimental results are as follows:

(1) Codebook size for mixed speech from English and Japanese is almost twice as large as the codebook size of either English or Japanese, but does not have to be as much as the codebook size of two speakers.

(2) Although many code vectors occurred in both English and Japanese, some code vectors have a tendency to predominantly occur in Japanese or in English.

(3) Code vectors which predominantly occurred in English are contained in /r/,/æ/,/f/,/š/, and code vectors which predominantly occurred in Japanese are contained in /i/,/u/,/N/.

(4) Judging from listening tests, English speech decoded by Japanese codebook can be also recognized as English.

Secondly, we proposed cross-language voice conversion methods based on a codebook mapping. The experiment results indicate that, because of the inconsistency of code vector correspondences and large spectrum differences between human speech and synthesized speech, the performance in cross-language voice conversion is less effective than in voice conversion between Japanese two speakers.

In this early stage of the project, we simply neglect code vectors which predominantly occurred in English or Japanese, and we used an 8-bit codebook. To improve the converted speech quality, we also have to increase the codebook size, and to estimate or extrapolate code vectors which predominantly occurred in English or Japanese. Because the cross-language voice conversion is a very new idea and also a very difficult problem, at this stage, we would like to say that we have at least shown the possibility of cross-language voice conversion and demonstrated possible methods.

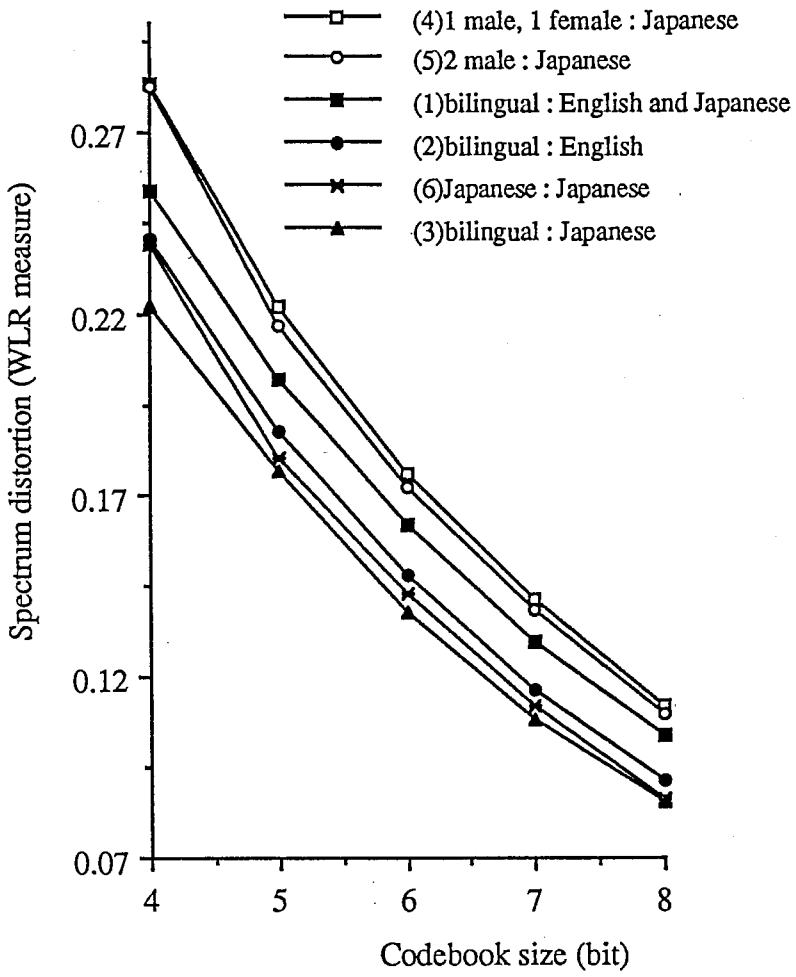


Fig. 4.1 Spectrum distortion for various codebooks

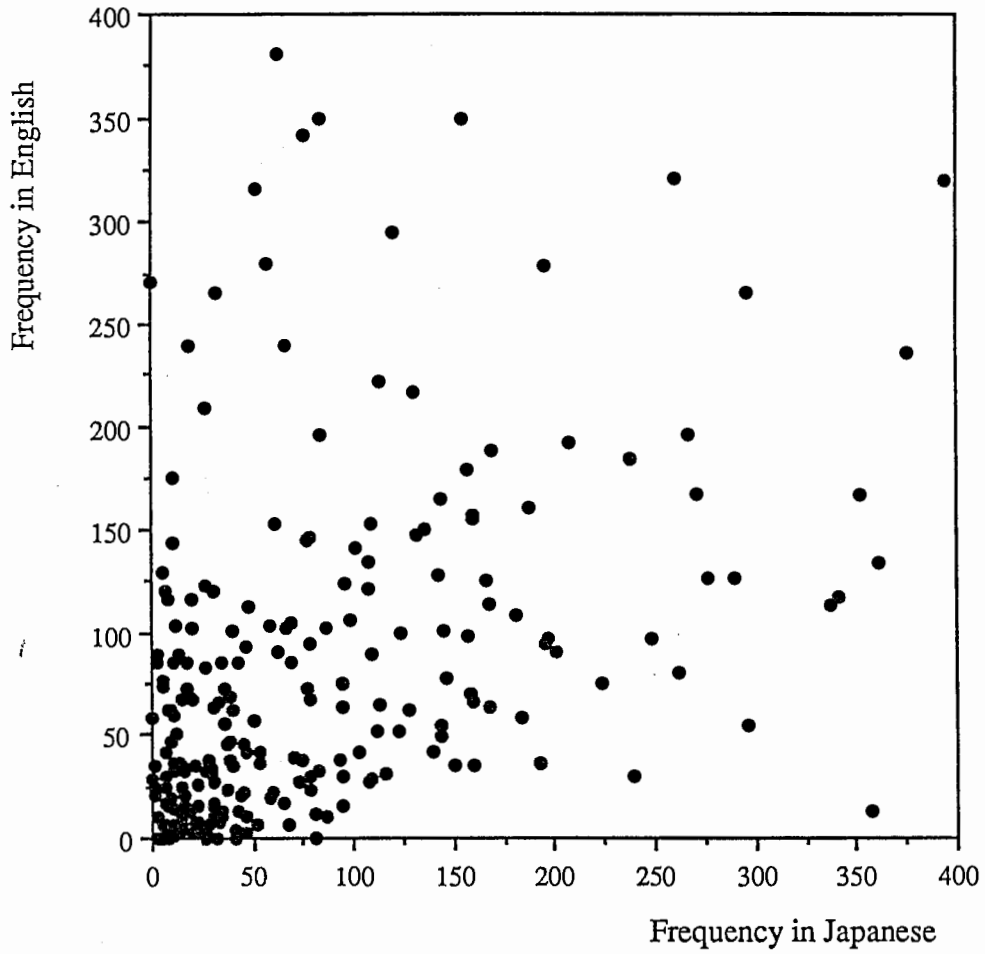


Fig. 4.2 Frequency of code vectors in a bilingual speaker's English and Japanese

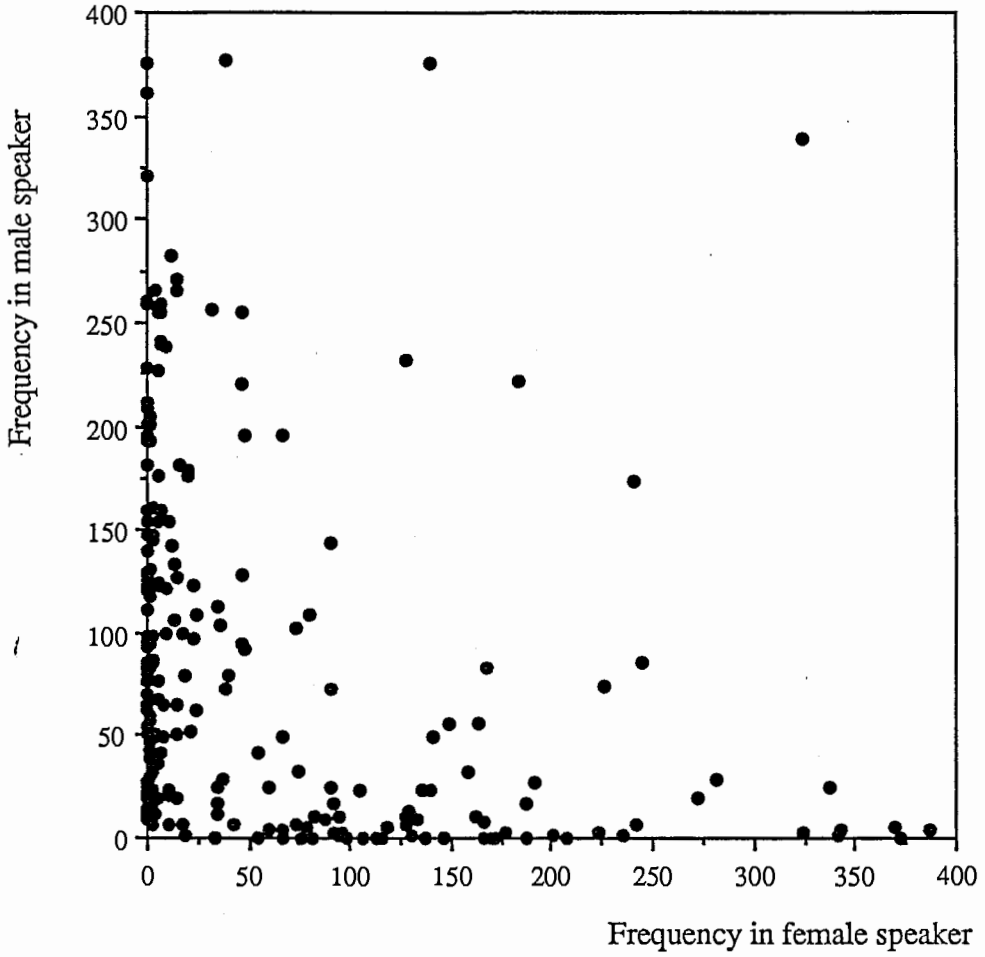
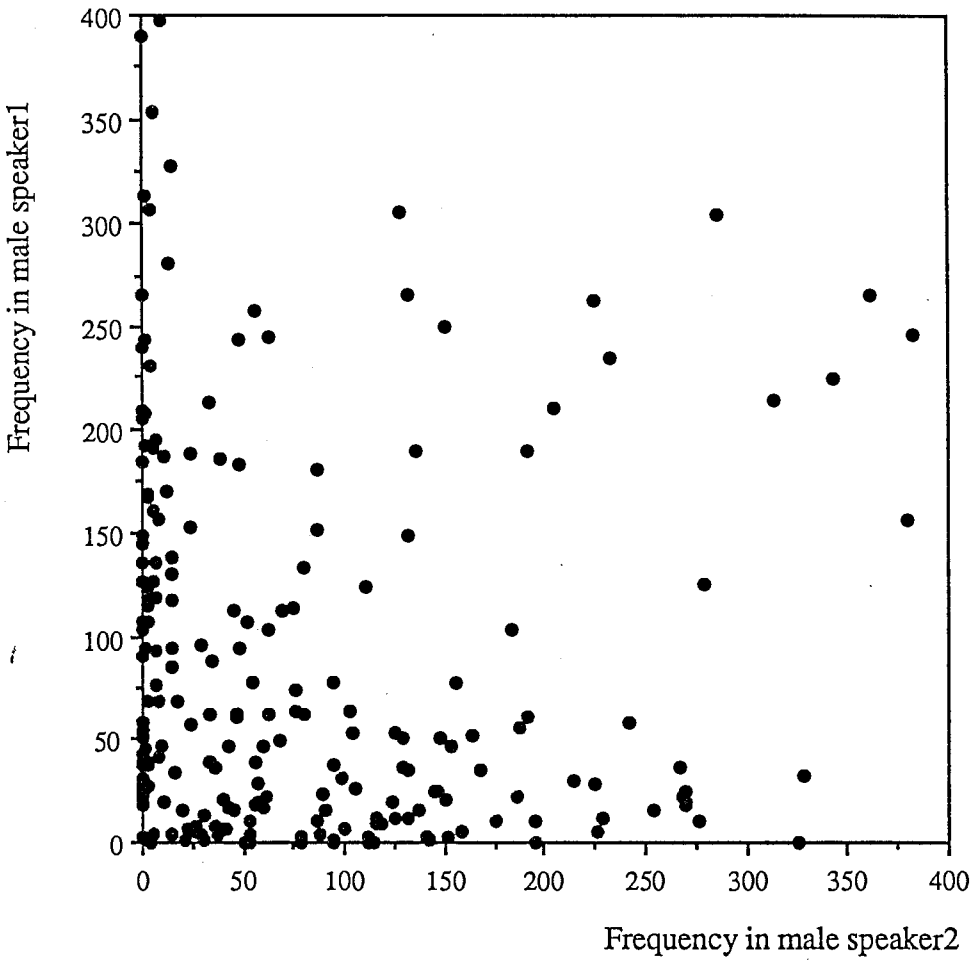


Fig. 4.3 Frequency of code vectors in male speaker and female speaker



F.ig. 4.4 Frequency of code vectors in different male speakers

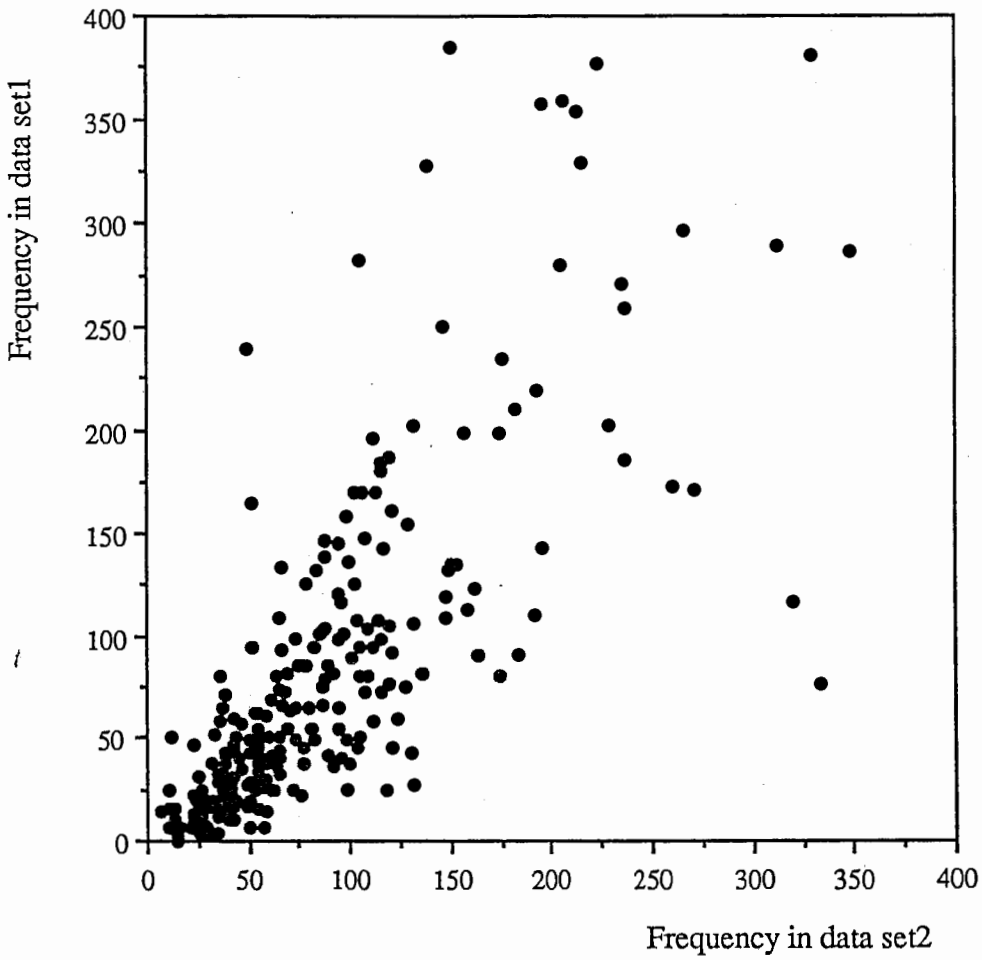
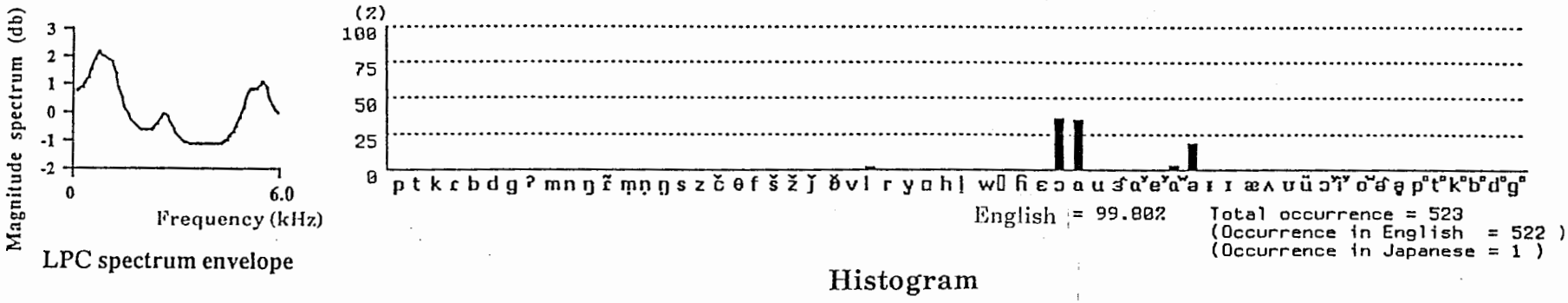
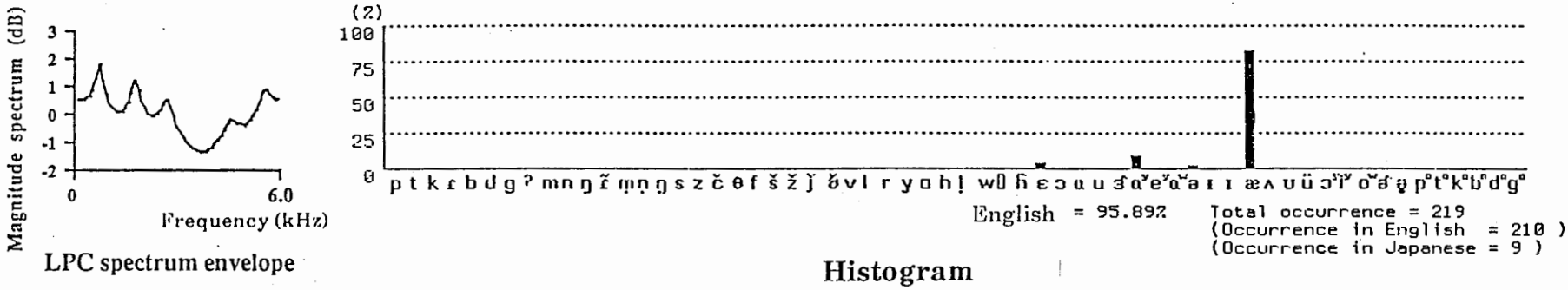


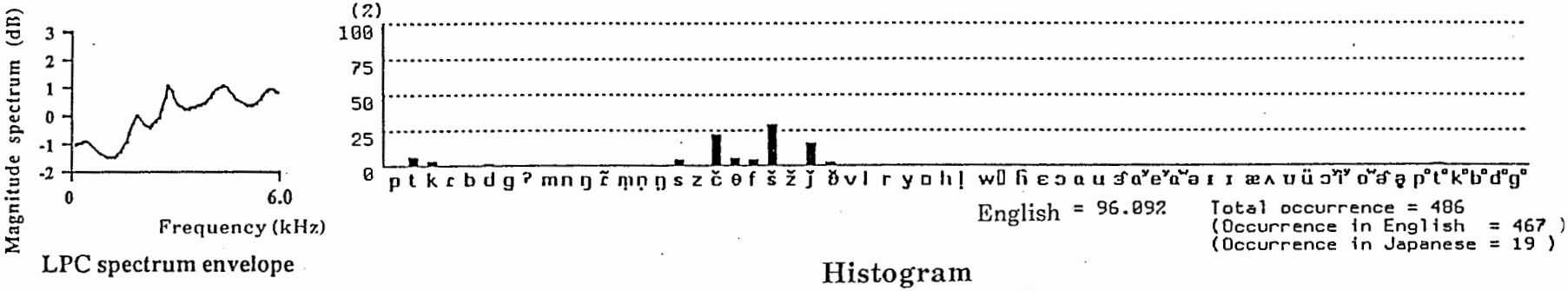
Fig.4.5 Frequency of code vectors in different data sets uttered by a same speaker



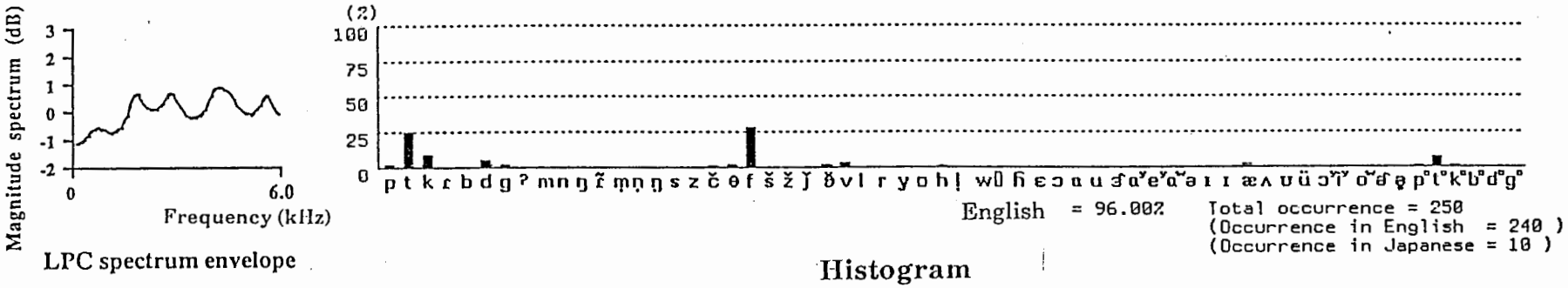
Flg. 4.6 LPC spectrum envelope and phoneme histogram in a code vector (A)



Flg. 4.7 LPC spectrum envelope and phoneme histogram in a code vector (B)



Flg. 4.8 LPC spectrum envelope and phoneme histogram in a code vector (C)



Flg. 4.9 LPC spectrum envelope and phoneme histogram in a code vector (D)

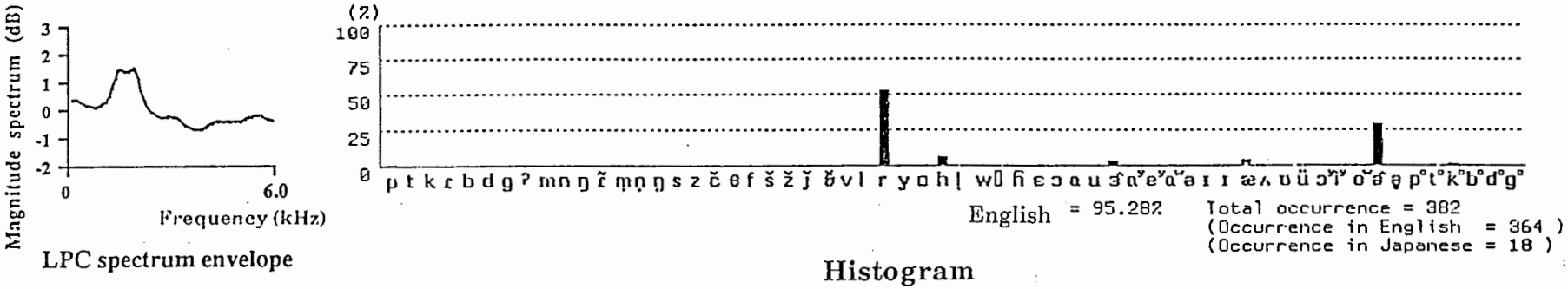


FIg. 4.10 LPC spectrum envelope and phoneme histogram in a code vector (E)

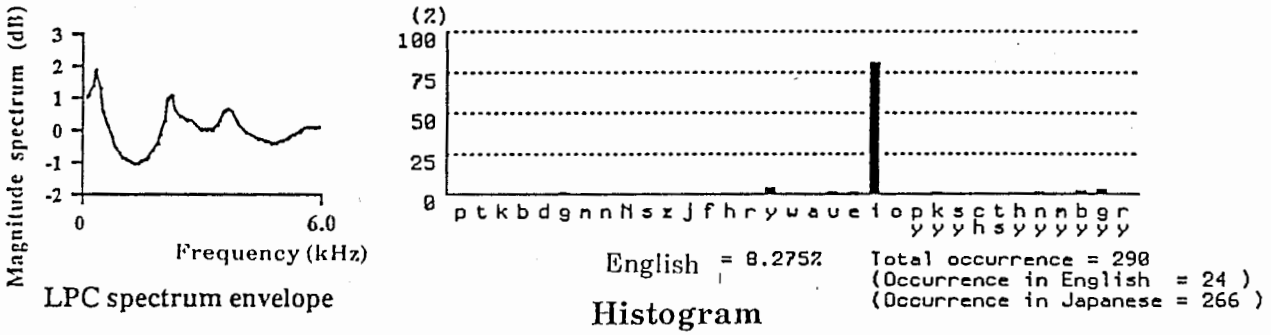
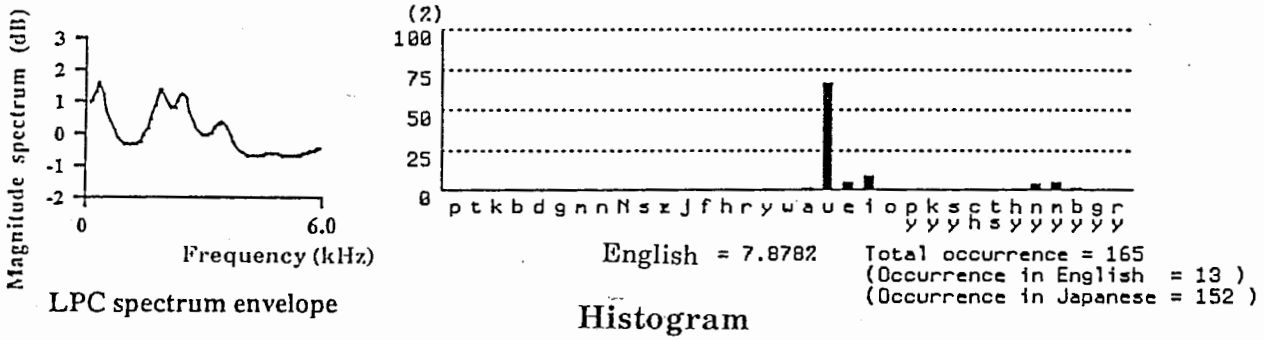
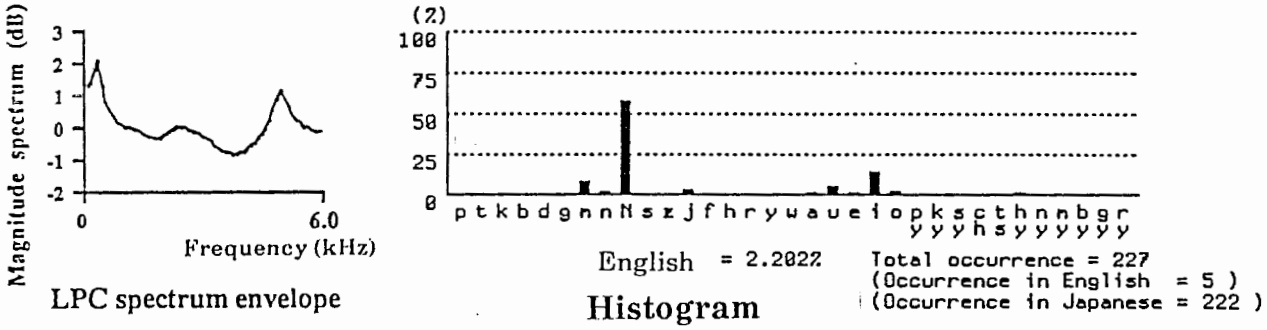


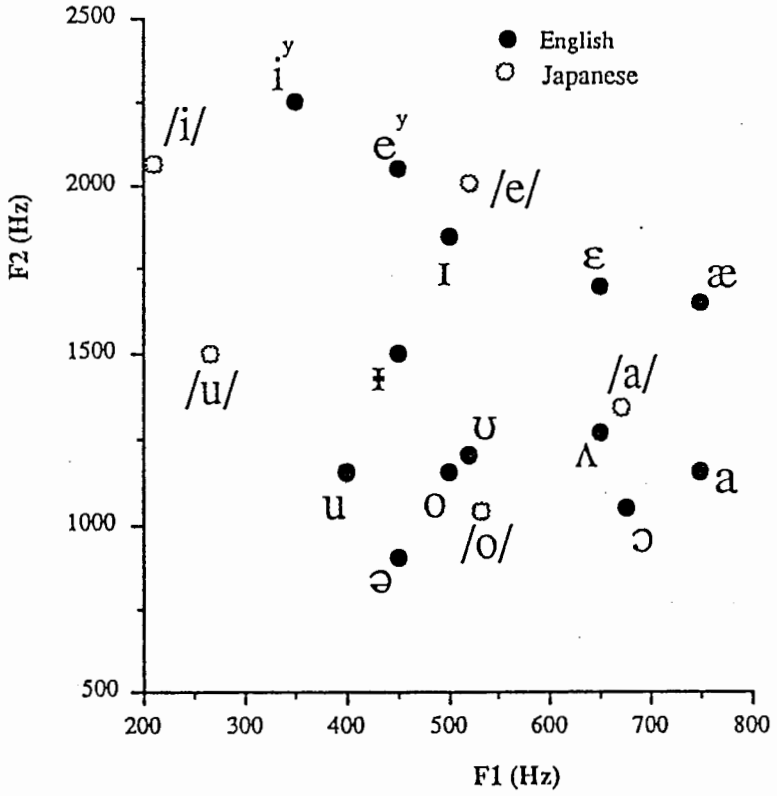
FIg. 4.11 LPC spectrum envelope and phoneme histogram in a code vector (F)



Flg. 4.12 LPC spectrum envelope and phoneme histogram in a code vector (G)



Flg. 4.13 LPC spectrum envelope and phoneme histogram in a code vector (H)



Flg. 4.14 F1 and F2 frequency in English and Japanese

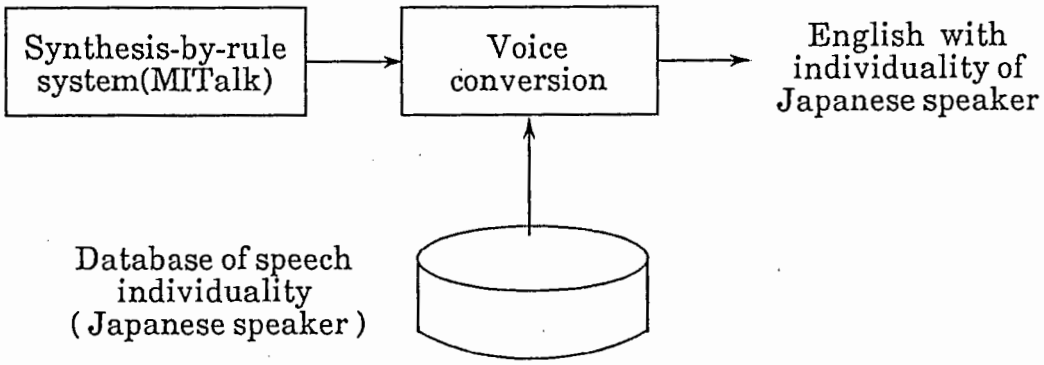


Fig.4.15 Cross-language voice conversion model

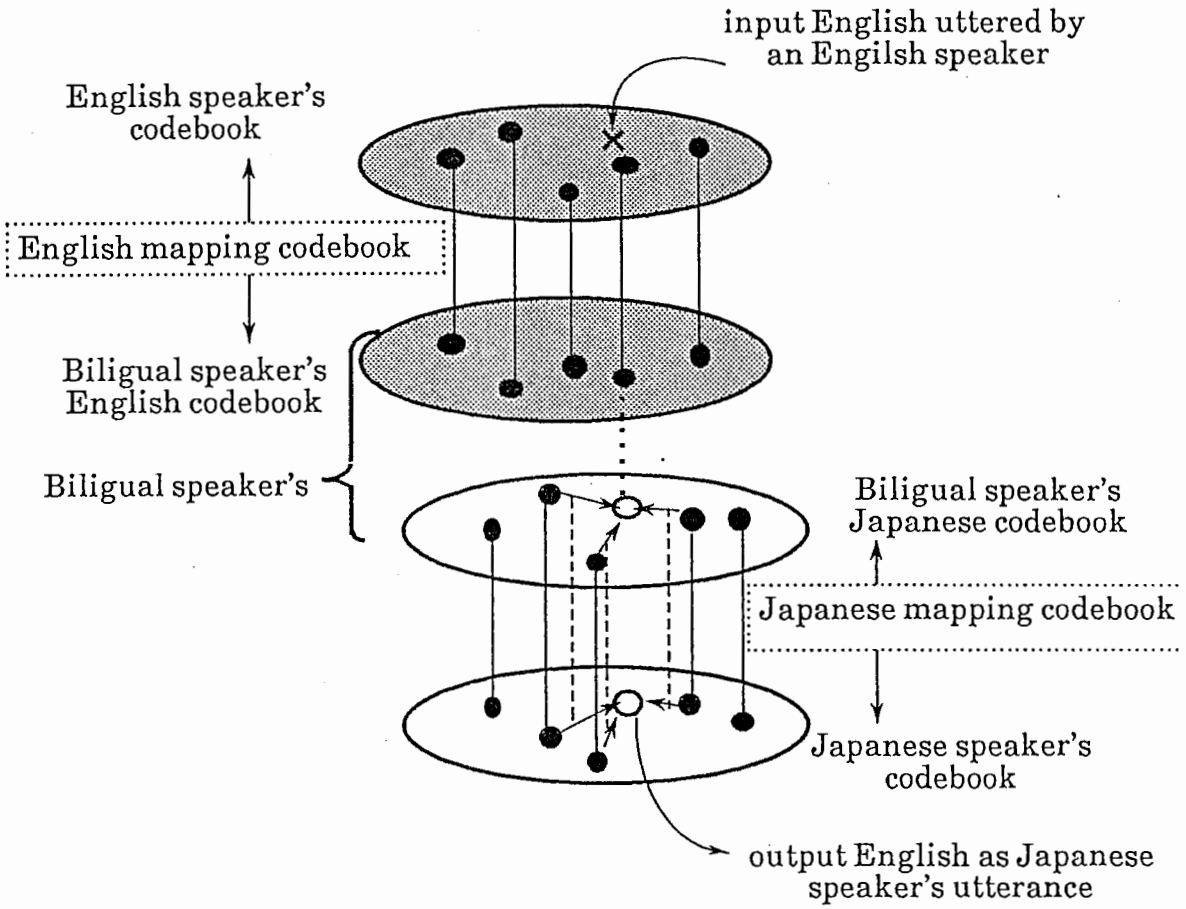


Fig.4.16 Cross-language voice conversion through a bilingual speaker

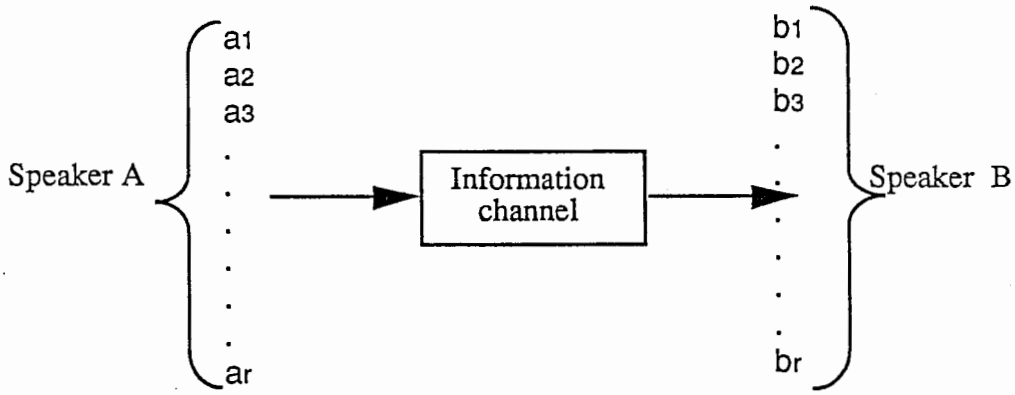
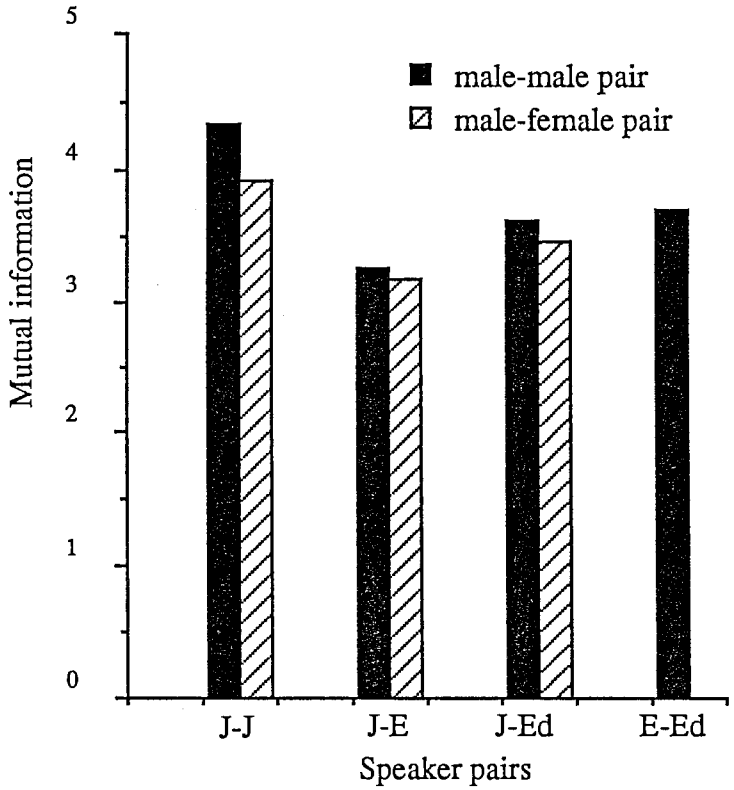
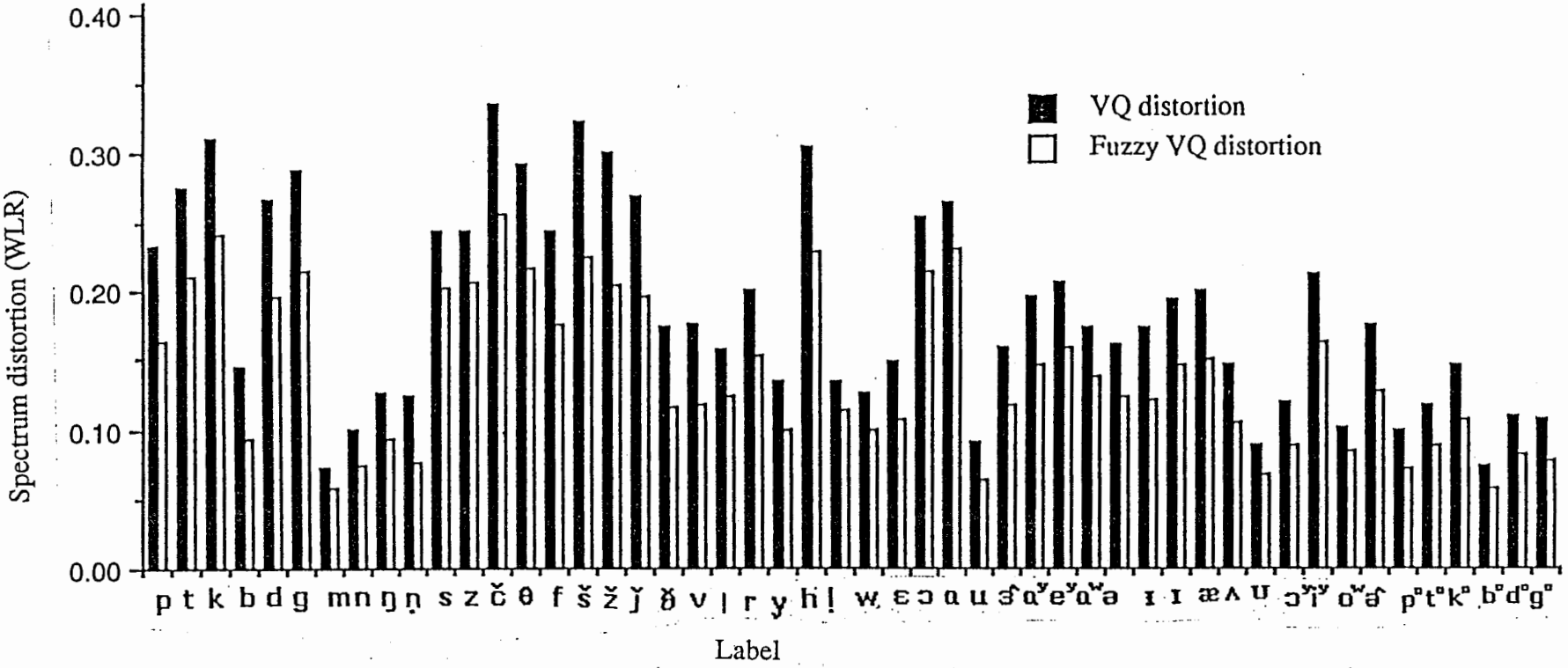


Fig. 4.17 Information channel aspect of voice conversion



J-J : Japanese speaker pairs
J-E: Japanese and MITalk-E pairs
J-Ed: Japanese and MITalk-Ed pairs
E-Ed: MITalk-E and MITalk-Ed pairs

Fig. 4.18 Mutual information for speaker pairs



Flg. 4.19 Spectrum distortion of English phonemes coded by Japanese codebook

Table 4.1 Experiment conditions

A/Ddata	12KHz sampling、16bit
window length	256points (21.3msec)
window shift	36points (3.0msec)
analysis order	14
clustering measure	WLR(Weighted Likelihood Ratio)
learning samples for clustering	12,000 frames
codebook size for spectrum parameter	256
learning words for mapping	100

Table 4.2 Kullback's divergence

Speech pair	Kullback's divergence
bilingual English vs. bilingual Japanese	1.21
male speaker vs. female speaker	8.59
male speaker1 vs. male speaker2	4.80
word set1 vs. word set2	0.21

Table 4.3 Correspondence between code vectors and phonemes

Code-vectors	Occurence in English (times)	Occurence in Japanese (times)	found in phoneme:
A	522	1	/ɔ/(35%), /ɑ/(34%), /ə/(23%), etc.(8%)
B	210	9	/æ/(78%), /ɑˇ/(10%), /ɛ/(7%), etc.(12%)
C	486	19	/š/(28%), /č/(24%), /ǰ/(20%), /ɵ/(10%), /f/(10%), etc.(8%)
D	240	10	/f/(28%), /t/(25%) /k/(12%), /d/(7%), /v/(6%), etc.(22%)
E	364	18	/r/(52%), /ʃ/(28%), /h/(8%0, etc.(12%)
F	24	266	/i/(78%), /j/(6%), etc.(16%)
G	13	152	/u/(73%), /i/(12%), /e/(7%), etc.(8%)
H	5	222	/N/(56%), /i/(20%), /n/(12%), /u/(8%), etc.(4%)

Table 4.4 Listening experiment result

judged correctly	judged incorrectly	indistinguishableble
27.2%	18.8%	54.0%

Table 4.5 Word category

judged correctly	judged incorrectly	indistinguishableble
noise, should, finger, outer, cashmere, masquerade, moisture, sculpture	personnel, with, zoologist, corsage, money	victor, fish, noteworthy, vocabulary, Irish, before, they, precaution, sweet, earthquake, hand, sweater, nothing, quite, ambiguous

Table 4.6 Spectrum distortion(WLR) in fuzzy VQ

codebook / coded utterance	Japanese		English
	VQ	fuzzy VQ	VQ
English	0.140	0.102	0.101

Table 4.7 Spectrum distortion(WLR) in fuzzy mapping

mapping codebook / coded utterance	Japanese		English	
	VQ	fuzzy VQ	VQ	fuzzy VQ
English	0.304	0.285	-	-
Japanese	-	-	0.285	0.268

Chapter 5

A Segment-Based Approach to Voice Conversion

1. Introduction

In chapter 2, we have already proposed a voice conversion algorithm based on codebook mapping and evaluated its performance. According to the mapping codebook, a speaker A's code vector is replaced frame-by-frame with a corresponding speaker B's code vector. The algorithm makes it possible to convert static characteristics(spectrum envelopes). However, dynamic characteristics can not be converted, because a code vector sequence which represents dynamic characteristics such as a formant trajectory, is obtained from speaker A's speech. This is one reason why voice conversion performance is not satisfactory.

In this chapter, to improve voice conversion performance, we propose to also convert the dynamic characteristics of speaker individuality by using speech segments as conversion units. Because speech segments contain both static and dynamic characteristics of speaker individuality, the use of segment units makes it possible to convert all of these parameters together. The advantages of a segment-based approach had been reported in the speech analysis-synthesis by HMM[Soong, 1989], segment vocoding[Shiraki, 1988][Peterson, 1990], and speech synthesis-by-rule systems[Nakajima, 1988][Sagisaka, 1988].

As the first step of the segment-based approach, we use phonemes as speech segment units. The reasons are as follows: (1)Because phonemes are distinctive features of sound in terms of speech perception and production, it is expected that speaker individuality is consistently preserved in phonemes. (2)In a segment-based approach, whether unit length is uniform or non-uniform is a matter of some concern. According to a recent study in segment vocoding[Peterson, 1990], non-uniform length units showed good performance and their length essentially depended on the phonemes. (3)To use non-uniform length segments requires expensive computation time. If phonemes are used as segment units, we can use speech recognition technologies making it possible to dramatically reduce search space for segment boundaries and segment units.

In section 2, we propose a voice conversion algorithm based on speech segment mapping. In section 3, the performance of the proposed algorithms are evaluated by measuring distortion and listening tests.

2. A Segment Based Approach to Voice Conversion

A voice conversion algorithm based on speech segment mapping is shown in Fig. 5.1. Both speakers had to have uttered the same training sentences or words in advance(hereinafter referred to as the speech database).

The algorithm consists of both off-line and on-line procedures. In the off-line procedures, a "correspondence table" which indicates the corresponding segments between the two speakers is generated, and Hidden Markov Models(HMM) are generated for a speaker A. Using the correspondence table and the HMM models, voice conversion is performed on-line as follows.

(1) Speech uttered by the speaker A is analyzed by LPC analysis.

(2) The input speech is mapped(recognized) into a sequence of phonemic symbols and segmented by the HMM models.

(3) A speech segment which optimally matches the segmented input speech is selected from the speaker A's speech database based on phonemic context constraints and a minimum distortion criterion.

(4) According to the correspondence table, the optimally selected speaker A's segment is replaced by the corresponding speaker B's segment.

(5) The pitch frequencies extracted from the input speech are linearly converted in order to match the pitch frequency range of the speaker B.

(6) Speech is synthesized by LPC synthesizer using the replaced (speaker B's) segments and the converted pitch frequencies.

In the following subsections, module specifications are described.

2.1 Correspondence Table Generation

Both speaker A and speaker B read a training corpus, but phonemic boundaries were only assigned manually to speaker A's speech. The time

alignment between speaker A's and speaker B's speech was obtained by dynamic time warping(DTW), and phonemic boundaries for speaker B's speech were assigned. The correspondence table contains phoneme symbols and phoneme segment IDs which correspond to the phonemic boundaries of speaker A and speaker B.

2.2 Segmentation Module

Speech segmentation was performed using the "Phonetic Typewriter" developed at ATR[Kawabata, 1990]. This system is not task specific, but is designed to map Japanese speech into a sequence of phonemic symbols based on the statistical model of phonemes and Japanese syllable occurrence. The system consists of an HMM recognition module and an LR parsing module. The total recognition system is shown in Fig. 5.2. Phonemic boundaries were determined by Viterbi algorithm.

2.2.1 HMM recognition module

The HMM model for five vowels, a syllabic nasal and a silence, was a 1-state model and the HMM model for the other phonemes, 39 in total, was a 3-state model. These phone units were trained using 5,557 isolated words uttered by the speaker A. The speech was transformed to VQ code sequence using 12th order LPC analysis and a 21.3 msec Hamming window with a 9msec frame shift. A multiple-codebook method was used, i.e., codebooks were separately generated for spectrum parameter, LPC cepstral difference, and power. HMM duration parameters were modified to match the speaking rate.

2.2.2 LR parsing module

The LR parser calculated the phone sequence probability based on syllable trigrams and HMM probabilities. The syllable trigram tables were made from a text database of over 35,000 syllables. The text database consisted of editorial

columns and transcribed texts from a telephone dialog simulation. An n-gram probability was interpolated from k(=0, 1, 2, 3)-gram probabilities by a "deleted interpolation" algorithm[Jelinek, 1980]. During a beam search, only 250 phone sequence candidates were maintained.

2.3 Optimal Segment Selection and Concatenation

Figure 5.3 shows an algorithm to select a speech segment which optimally matches the segmented input speech. Phoneme segments were first selected based on triphone context constraints using recognized output symbols. If there was no phoneme in the database under the triphone context, the current phonemes were all candidates.

Based on a minimum distortion criterion, an optimal phoneme segment was selected by DTW. Cepstrum distance measure was used. If there was no segment because of the DTW path constraint, DTW was again performed after uniformly lengthening or shortening the segmented input speech.

Finally, the selected phoneme segment was concatenated to the next phoneme segment in a frame which gave the minimum spectrum distortion. The final procedure was necessary to adjust inadequate segment boundaries introduced by the correspondence table generation and the HMM segmentation.

3. Performance Evaluation

The proposed voice conversion algorithm was performed between two male speakers, i.e., speaker A's speech was converted to sound like speaker B's speech.

In the off-line procedures, a correspondence table and HMM models for the speaker A are generated using isolated word utterances. In the on-line procedures, twenty-five continuously uttered sentences were converted. The

sentences, which contained 279 phrases, were collected through simulation of a secretarial service for an international conference.

In section 3.1, a correspondence table specification is shown. In section 3.2, segmentation module performance is shown. Voice conversion performance is evaluated by spectrum distortion and listening test in sections 3.3 and 3.4, respectively. In sections 3.3 and 3.4, only correctly recognized phrases were evaluated. Cases where recognition errors occurred in the segmentation module are discussed in section 3.5.

3.1 Correspondence Table

Both speaker A and speaker B read a training corpus of 1,323 Japanese words. The speech was sampled at 12 kHz and analyzed using a 14th order LPC analysis. The analysis window was a 21.3 msec Hamming window with a 3.0 msec frame shift.

Table 5.1 shows the phoneme symbols, phoneme segment numbers and average phoneme durations contained in the speech databases of both speaker A and speaker B.

The phonemic boundaries which were given to speaker B's speech according to the DTW path were compared with manually assigned segment boundaries of the speech. The average boundary errors and coincidence rate within a 20-msec time window are also shown in Table 5.1. Boundary assignment performance is relatively poor in the /r/, /hy/, /f/, /u/, /sh/, /w/, /ry/ phonemes, but for the other phonemes, more than 70% of the boundaries are assigned within 20 msec of the manual segmentation point.

3.2 Recognition and Segmentation Performance

The phone recognition rate is 94.9% and the phrase recognition rates of the top choice and the top 5 choices are 78.5% and 92.1%, respectively[Kawabata, 1990].

The automatic segmentation results are depicted in Fig. 5.4 in terms of their coincidence rate with manual segmentation boundaries, where only the correctly recognized phrases were adopted. The coincidence rates are defined as the percentage of phonemes for which the automatic segmentation and the manual segmentation boundaries agree a)within a 15 msec window and b)within a 20 msec window. Boundary assignment performance is relatively poor in voiced fricative /z/, and voiced affricate/j/, but for the other phonemes more than 80% of the boundaries are set within a 20 msec range.

3.3 Voice Conversion Evaluation by Spectrum Distortion

3.3.1 Experimental Procedure

The voice conversion performance was evaluated by spectrum distortion. By way of comparison, both manually segmented continuous speech utterances and manually segmented isolated word utterances(88 words) are also converted in addition to automatically segmented continuous speech utterance.

Spectrum distortion is calculated by DTW using cepstrum distance measure for the following data.

(1) Speech uttered twice by the speaker A(hereinafter referred to as the 1st utterance and the 2nd utterance)

(2) The 1st utterance and a speech segment sequence which was selected optimally to match the 1st utterance from speaker A's database(hereinafter referred to as segment vocoder)

(3) The 2nd utterance and segment vocoder

(4) Speech uttered by speaker B and speech converted by the proposed algorithm(hereinafter referred to as segment conversion)

(5) Speech uttered by the speaker B and speech converted by the codebook mapping frame-by-frame(hereinafter referred to as VQ conversion)

(6) The 1st utterance of speaker A and speech uttered by the speaker B

VQ conversion was performed by the codebook mapping algorithm. For a fair comparison of VQ conversion and segment conversion, a mapping codebook(10bits) was trained using 1,300 words.

3.3.2 Experimental Results

Figures 5.5 and 5.6 show the experimental results using isolated utterances and continuous utterances, respectively. In both isolated and continuous utterance, the distortion caused by utterance times is small, 0.2481 and 0.2623 respectively. This is considered to be the goal of the voice conversion.

In terms of the distortion between the 1st utterance and the segment vocoder, the distortion in continuous speech(0.3755) is much higher than the distortion in isolated utterances(0.2832), because speaker A's database consisted of isolated word utterances.

In terms of the distortion between the 2nd utterance and the segment vocoder, the distortion in manually segmented speech(0.3638) is much less than the distortion in automatically segmented speech(0.4471).

Taking into account the distortion caused by utterance times, speaking styles and automatic segmentation, the distortion goal of this experiment is the distortion between the 2nd utterance and the segment vocoder, 0.4471. The natural distortion between speaker A and speaker B is 0.9396. The distortion between the segment conversion and the target speech is 0.6169. Therefore, the segment conversion reduces the distortion by one-third.

When manually segmented speech is used instead of automatically segmented speech, the distortion between the segment conversion and the target speech is 0.5369, which is less than the distortion between the VQ conversion and the target speech. The results indicates that segment-based approach has more potential than the frame-wise approach.

The distortion between the 1st utterance and the segment vocoder should be less than the distortion between the 2nd utterance and the segment vocoder. However, the experimental results show just the opposite. This implies that segment variations in the speech database are not enough.

Judging from these results, to get higher voice conversion performance, it is important to improve segmentation performance and to add continuous speech to the speech database.

3.4 Voice Conversion Evaluation by Listening Test

3.4.1 Experimental procedure

3.4.1.1 Experiment 1

In Experiment 1, three tests(test 1.1, test 1.2, and test 1.3) were carried out by the ABX method. Stimuli A and B were LPC analysis-synthesis of speaker A's or speaker B's speech. In test 1.1, X was speech from the segment conversion. In test 1.2, X was speech from the VQ conversion. In test 1.3, X was speech from the LPC analysis-synthesis speech of speaker B. Different phrase tokens were used for stimuli A, B and X, and all possible ABX combinations were generated. The ABX triads, 36 in total, were presented to twelve listeners using a headphone. The listeners were required to select the stimulus (A or B) which most closely resembled stimulus X in speaker identity.

3.4.1.2 Experiment 2

Experiment 2 was designed to evaluate the voice quality by a pair-comparison listening test. The following five different types of speech were synthesized as stimuli.

(1) LPC analysis-synthesis speech of speaker A(A-LPC)

(2) The segment vocoder(A-Segment)

(3) The VQ conversion(A→B-VQ)

(4) The segment conversion(A→B-Segment)

(5) LPC analysis-synthesis speech of speaker B(B-LPC)

Two different words were used to make speech pairs. A set of speech pairs, 40 in total, include all possible combinations of stimuli from the five different types of speech. The listeners heard the speech pairs under the same listening conditions as in Experiment 1, and were asked to rate the similarity for each pair using five categories: "similar","somewhat similar","difficult to decide","slightly dissimilar", and "dissimilar".

3.4.2 Experimental results

3.4.2.1 Results of Experiment 1

The result of Experiment 1 is shown in Table 5.2. The numbers in this table represent the percentage of responses where stimuli X was judged to be close to the LPC analysis-synthesis speech of speaker B.

Even though stimuli X is the LPC analysis-synthesis speech of speaker B, listeners misjudged 3.8 percent of the time. Judging from this, the speaker identification accuracy(93.8%) obtained by the segment conversion is quite high. Segment conversion performance is about 20% higher than the VQ conversion performance.

3.4.2.2 Results of Experiment 2

Hayashi's fourth method of quantification[Hayashi, 1985] was applied to the experimental data of Experiment 2. This method places stimuli on a space according to the similarities between any two stimuli, and its formulation minimizes the measure Q, where

$$Q = -\sum_{i=1}^{n}\sum_{j=1}^{n} e(i,j)\{x(i)-x(j)\}^2 \qquad (i \neq j)$$

$e(i,j)$ denotes the similarity between stimuli i and j, $x(i)$ represents the location of stimulus i in the space, and n is the number of stimuli.

The projection onto a two-dimensional space is shown in Fig. 5.7. It represents the relative similarity-distance between stimuli. Contribution rates which

indicate the importance of the axis are 60% and 30% for axis I and axis II, respectively. In terms of axis I values, the converted speech by both VQ conversion and segment conversion are very close to the target B-LPC. Therefore, axis I probably represents the speech individuality. On the other hand, the axis II values of VQ conversion, segment conversion, and segment vocoder are minus values, and the axis II values of analysis-synthesis speech are plus values. Therefore, axis II probably represents distortion caused by modifications. These results indicate that speech individuality is well converted, but also that the modification introduces some artificial noise to the converted speech.

3.5 Analysis of Recognition Error

In 60 phrases, recognition errors occurred. Problems introduced by recognition errors were examined by listening to LPC analysis-synthesis speech and segment vocoding speech of speaker A

In 5 phrases, recognition errors resulted in no sound problems in the segment vocoding speech. The errors are shown in table 5.3.

In 10 phrases, the segment vocoding speech did not sound perfect , but was relatively close to the LPC analysis-synthesis speech. In Fig. 5.8(a), 5.8(b), 5.8(c), spectrograms of the LPC analysis-synthesis speech and segment vocoding speech are shown. It is observed that they show relatively close patterns.

In the other phrases, 45 in total, recognition errors resulted in phonemes different from the input speech. Figure 5.8(d) shows where recognition errors occurred most frequently. The reasons recognition errors resulted in different phonemes are (1)in the optimal segment selection, recognized symbols are used to reduce search space, (2)in the database, there were not enough ambiguous phoneme pairs.

4. Conclusion

In this chapter, by introducing a segment based approach, we proposed a new voice conversion algorithm which makes it possible to control not only the static characteristics but also the dynamic characteristics of speaker individuality.

The proposed voice conversion algorithm was performed between two male speakers using 25 sentences which contain 279 phrases. The phone recognition rate was 94.9% and the phrase recognition rate of the top choice was 92.1%. Spectrum distortion between the target speech and the converted speech was reduced to one-third the natural spectrum distortion between the two speakers. A listening evaluation showed that, in terms of speaker identification accuracy, the speech converted by segment units gave a score 20 % higher than the speech converted frame-by-frame. We conclude that speech segments contain more information to represent speaker individuality than frames, and that the information difference between segments and frames is large enough to hear.

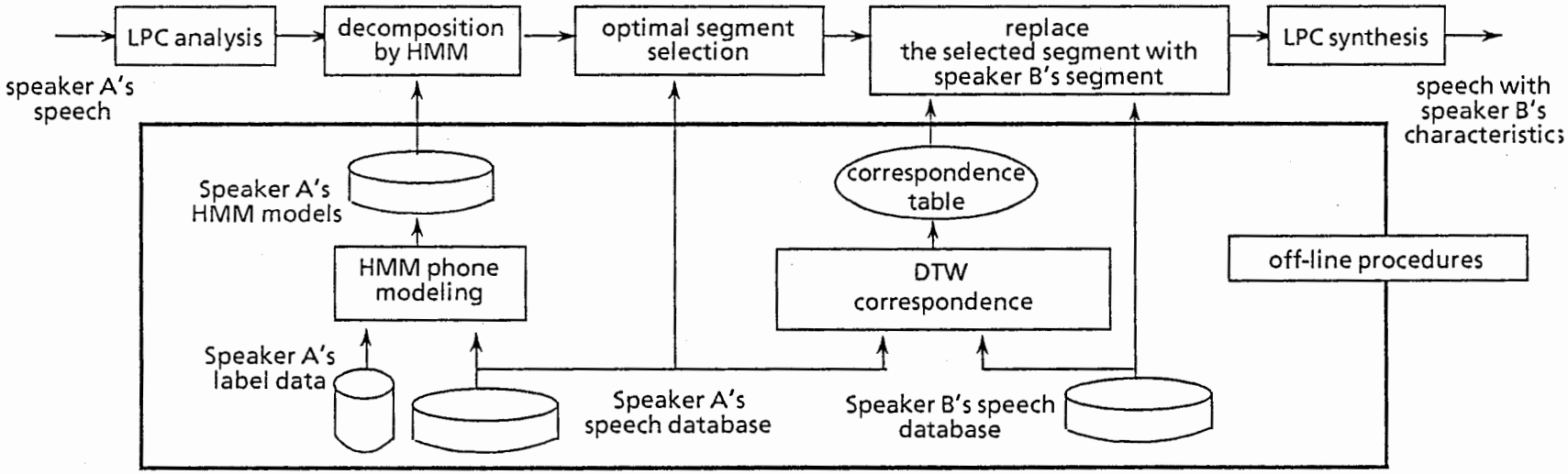


Fig. 5.1 Block diagram of segment-based voice conversion

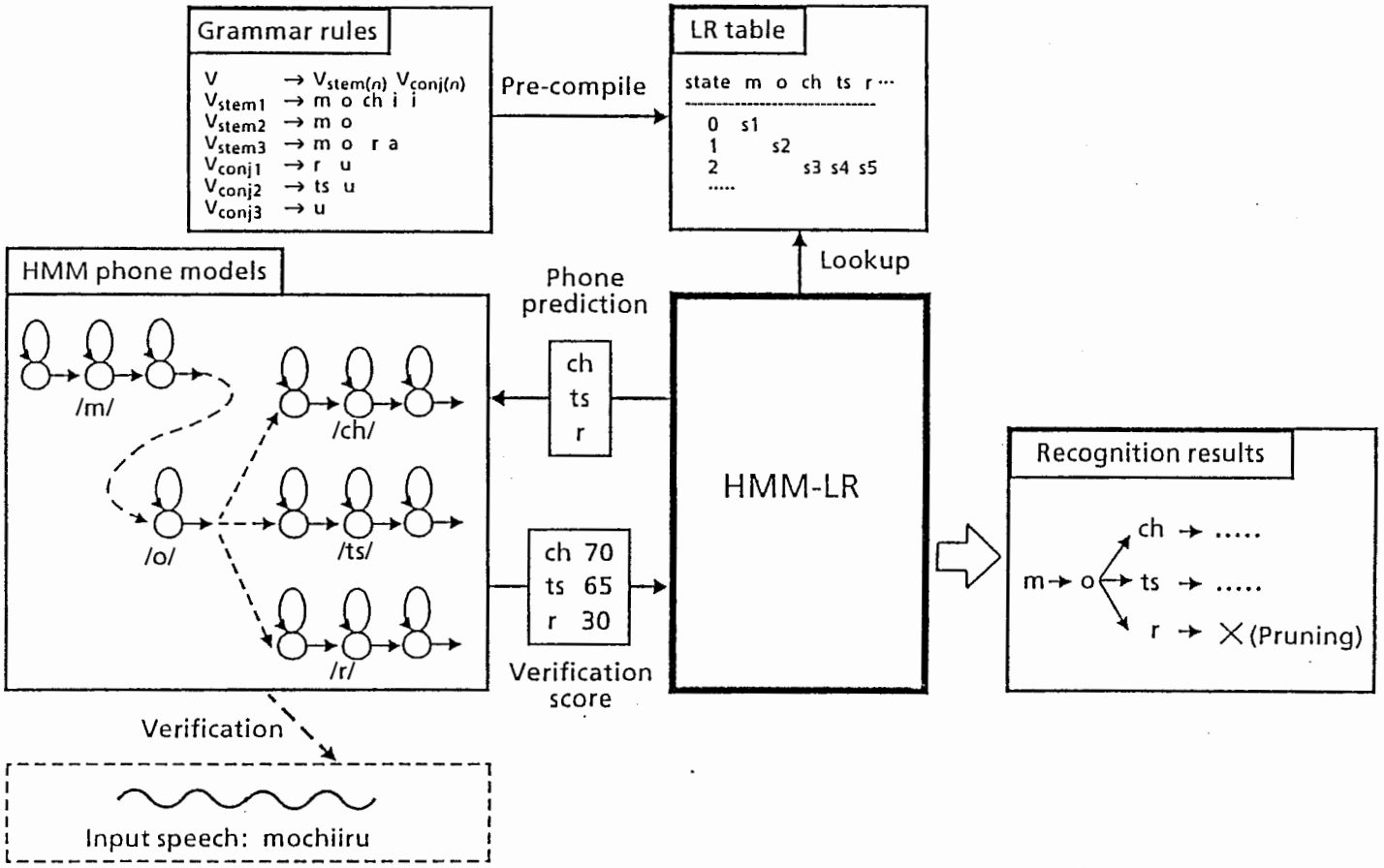


Fig.5.2 HMM-LR recognition system

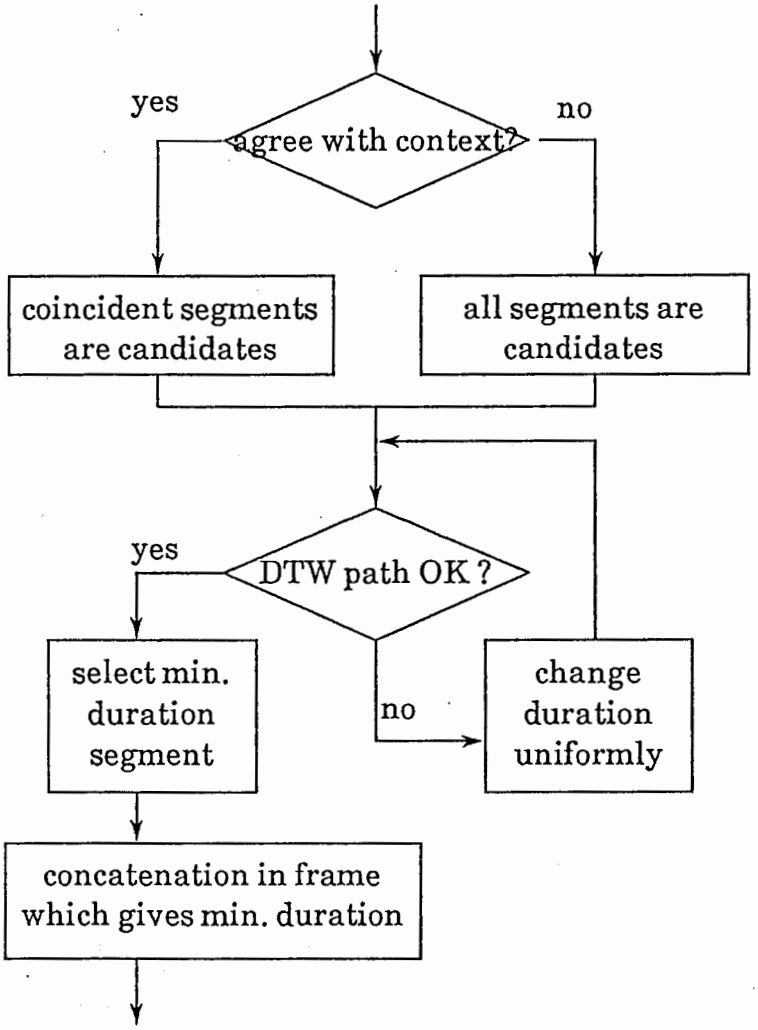
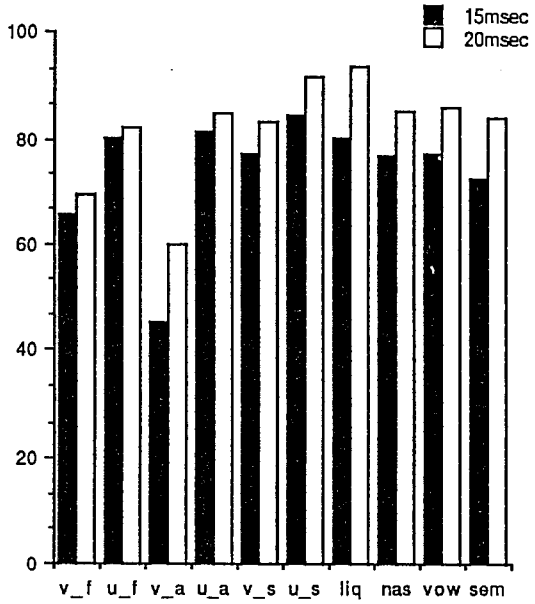


Fig.5.3 Optimal segment selection algorithm



v__f:voiced fricatives
u__f:unvoiced fricatives
v__a:voiced affricates
u__a:unvoiced affricates
v__s:voiced stops
u__s:unvoiced stops
liq:liquid
nas:nasals
vow:vowels
sem:semivowels

Fig.5.4 Segmentaion performance

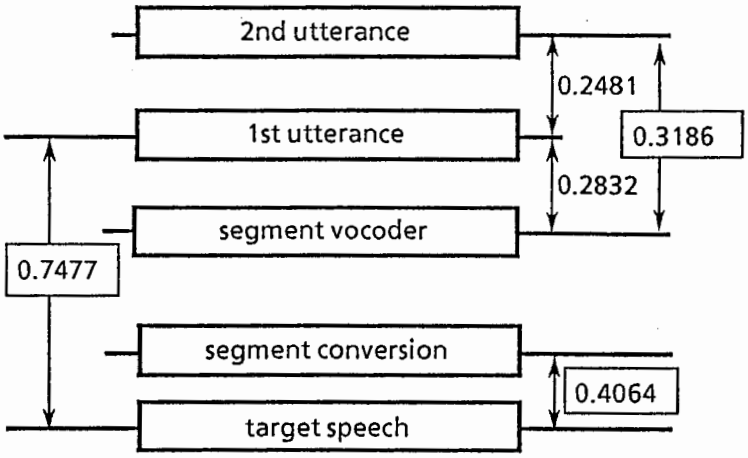
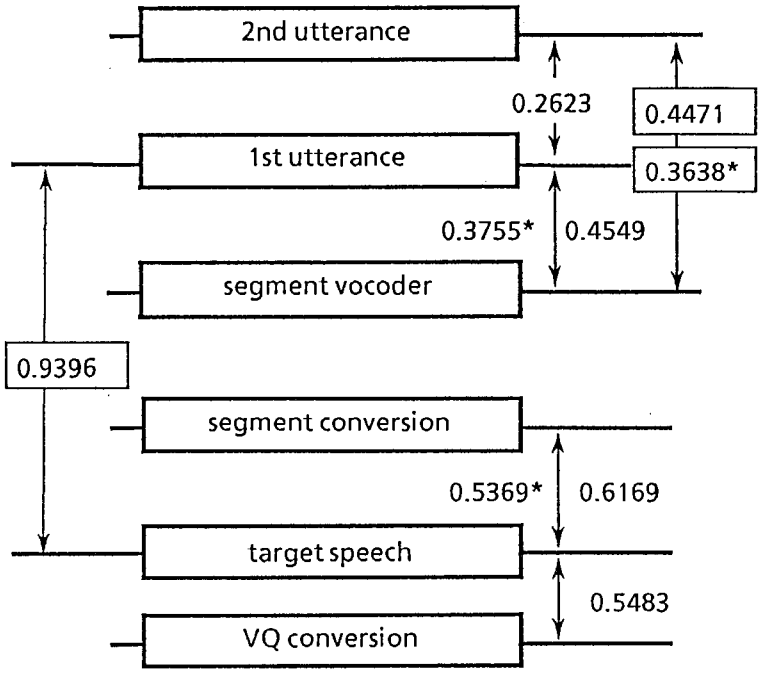
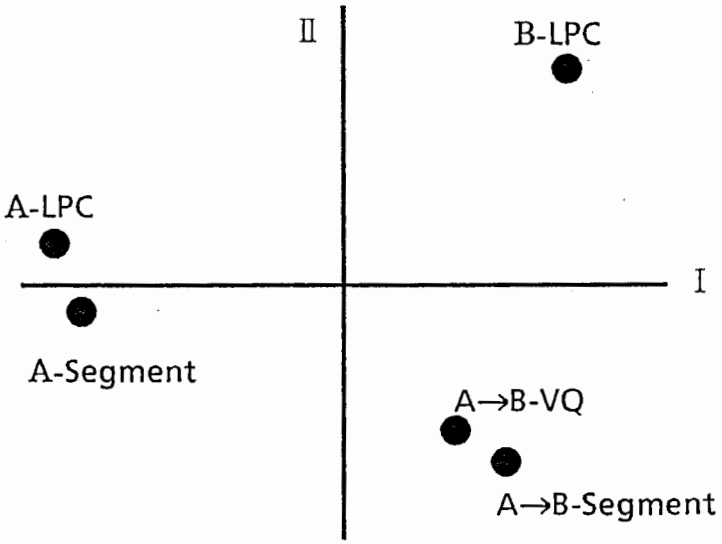


Fig. 5.5 Spectrum distortion in isolated utterances



* indicates manual segmention.

Fig. 5.6 Spectrum distortion in countinuous utterances



A-LPC: LPC analysis-synthesis speech of speakerA
A-Segment: segment vocoder
A→B-VQ: VQ conversion
A→B-Segment: segment conversion
B-LPC: LPC analysis-synthesis speech of the speakerB

Fig. 5.7 Distribution of psychological distances for the voice conversion

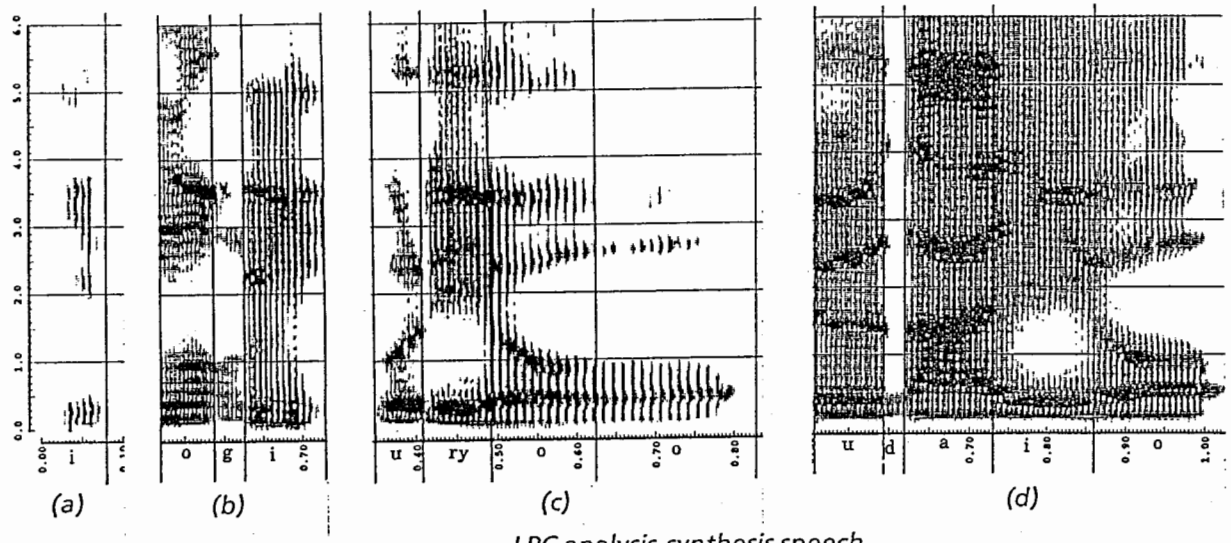


Fig. 5.8 Spectrogram of synthesized speech

Table 5.1 Segments in the database

symbols	number of segments	average duration (msec)	boundary errors (msec)	coincidence rate (%)
p	22	84.0	6.42	95.5
t	229	51.0	11.79	83.4
k	576	72.0	12.60	81.1
b	130	63.0	8.19	90.0
d	105	60.0	5.16	95.2
g	163	60.0	13.4	77.3
m	339	63.0	6.60	94.7
n	195	54.0	6.36	93.3
N	227	189.0	15.93	71.8
s	241	129.0	8.04	92.9
sh	115	162.0	19.8	64.3
ch	51	114.0	9.39	86.3
ts	97	111.0	12.66	80.4
z	69	75.0	14.07	79.7
j	78	108.0	15.57	76.9
f	22	78.0	26.85	54.5
h	130	66.0	16.47	76.9
r	547	21.0	33.57	42.4
y	75	72.0	9.00	89.3
w	55	54.0	15.15	69.1
i	783	120	15.69	75.0
e	404	126.0	16.35	76.7
a	874	111.0	10.26	85.5
o	578	111.0	8.55	90.0
u	852	117.0	23.34	66.6
gy	1	135.0	9.00	100
hy	1	117.0	39.00	0
ky	7	120.0	9.00	85.7
ry	8	93.0	19.11	62.5

Table 5.2 Percentages of correct responses

LPC analysis-synthesis of speakerB	segment conversion	VQ conversion
96.2%	93.6%	71.5%

Table 5.3 Recognition errors

Input	/ei/	/ou/	/ee/
output	/ee/	/oo/	/ei/

Chapter 6

Epilogue

In this thesis, we discussed algorithms to change speaker individuality: i.e., speech uttered by a speaker is changed to sound as if another speaker had uttered it.

In chapter 2, we formulated voice conversion as a mapping problem by introducing vector quantization. The advantage of this technique are summarized as follows:

(1) The mapping codebooks which make it possible to impart individuality to synthesized speech are generated from a limited number of word utterances.

(2) The mapping codebooks enable voice conversion between any two speakers.

(3) The synthesis process requires minimal computation and produces speech in real time.

The performance of this technique was confirmed by spectrum distortion and pitch frequency difference. The spectrum distortion between original speech and target speech decreased by a range of 27% to 66%. Pitch frequency difference decreased to less than 15Hz. The overall performance of this technique was also confirmed by listening tests. It can be concluded that the converted speech has a voice quality very close to the target speaker's.

To improve the naturalness and clarity of the converted speech, the usage of fuzzy VQ and difference vectors was discussed. According to the listening test, fuzzy VQ improved smoothness by generating spectrum patterns beyond the limitations imposed by the codebook size, and the usage of difference vectors was quite effective in improving clarity by representing spectrum characteristic details ignored by VQ or fuzzy VQ.

In chapter 3, we proposed a new algorithm which makes it possible to synthesize high quality speech even if pitch frequency or duration is somewhat changed. The advantages of this algorithm are listed below;

(1) This algorithm needs no phase unwrapping which is the most complex and critical procedure in the conventional method.

(2) This algorithm is easy to implement in an automatic system because explicit pitch frequency extraction is not required.

(3) The quality of synthesized speech is very high and natural because residual signals are used as excitation.

(4) This algorithm makes it possible to modify the spectrum envelope in a non-parametric way because it is represented by FFT magnitude.

The listening test showed that the proposed algorithm was able to reproduce high quality speech sounds even if the pitch frequency was modified both in a uniform and a non-uniform manner.

In chapter 4, speaker individuality control across different languages was discussed. To apply a voice conversion algorithm based on codebook mapping to the cross-language voice conversion, speech uttered by a bilingual speaker was analyzed. Experimental results are as follows:

(1) The codebook size for mixed speech from English and Japanese is almost twice as large as the codebook size of either English or Japanese, but does not have to be as large as the codebook size of two speakers.

(2) Although many code vectors occurred in both English and Japanese, some code vectors have a tendency to predominantly occur in Japanese or in English.

(3) The code vectors which predominantly occurred in English are contained in /r/,/æ/,/f/,/š/, and the code vectors which predominantly occurred in Japanese are contained in /i/,/u/,/N/.

(4) Judging from listening tests, English speech decoded by Japanese codebook can be also recognized as English.

We proposed cross-language voice conversion methods based on codebook mapping. The experiment results indicated that, because of the inconsistency in code vector correspondences and the large spectrum differences between human

speech and synthesized speech, the performance in cross-language voice conversion was less effective than in voice conversion between two Japanese speakers.

In paper 5, by introducing a segment based approach, we proposed a new voice conversion algorithm which makes it possible to control not only the static characteristics but also the dynamic characteristics of speaker individuality.

The proposed voice conversion algorithm was performed between two male speakers using 25 sentences which contain 279 phrases. The phone recognition rate was 94.9% and the phrase recognition rates of the top choice and the top 5 choices were 78.5% and 92.1%, respectively. Voice conversion was evaluated using correctly recognized phrases. Spectrum distortion between the target speech and the converted speech was reduced to one-third the natural spectrum distortion between the two speakers. Listening evaluation showed that, in terms of speaker identification accuracy, the speech converted by segment units gave a score 20 % higher than the speech converted frame-by-frame. We conclude that speech segments contain more information to represent speaker individuality than frames, and the information difference between segments and frames is large enough to hear.

In summary, we have confirmed that the speaker individuality control problem is successfully formalized as a mapping problem. As a future work, we would like to expand this approach to control other kinds of speaker individuality, such as source characteristics, speaking styles and dialects.

Bibliography

[Abe, 1988] M. Abe, S. Nakamura, K. Shikano, H. Kuwabara, "Voice Conversion Through Vector quantization", ICASSP88, 655-658, 1988.

[Abe, 1988b] M. Abe, S. Tamura, H. Kuwabara, "A Speech Modification Method by Signal Reconstruction Using Short-Time Fourier Transform", IEICE (D-Ⅱ)、Vol. J72-D-Ⅱ, No.8, pp.1180-1186, in Japanese.

[Abe, 1990a] Abe, M., Shikano, K., Kuwabara, H.."Cross-language Voice Conversion," ICASSP90, 345-348.

[Abe, 1990b] Abe, M., Shikano, K., Kuwabara, H.. "Voice conversion for an interpreting telephone," ESCA Proc. Workshop on Speaker Characterization in Speech Technology, 40-45.

[Abe, 1990c] M.Abe, S.Nakamura, K.Shikano, H.Kuwabara, "Voice conversion through vector quantization", The Journal of the Acoustical Society of Japan 、Vol. 11, No.2, pp.71-76.

[Allen, 1979] Allen, J, Hunnicutt S. "MITalk-79: The 1979 MIT Text-to-speech system", Proc. of the 97th Meeting of ASA.

[Buzo, 1980] A. Buzo, A. H. Gray, Jr., R. M. Gray, J. D. Markel, "Speech coding based upon vector quantization," IEEE, ASSP, Vol. ASSP-28, No. 5, pp. 562-674, Oct. 1980.

[Charpentier, 1986] F.J.Charpentier, M.G.Stella, "Diphone Synthesis Using an Overlap-Add Technique for Speech Waveforms Concatenation," ICASSP'86

[Childers, 1985] D. G. Childers, B. Yegnanarayana, K. Wu, "Voice Conversion: Factors Responsible for Quality", ICASSP85, 748-751, 1985.

BIBLIOGRAPHY

[Flanagan 1972] J. L. Flanagan, "Speech analysis synthesis and perception", 2nd ed., Springer Verlag, Berlin, Heidelberg, New York, 1972.

[Furui, 1981] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features", IEEE ASSP-29, 3, 342-350, 1981.

[Furui, 1985] S. Furui, M. Akagi, "Perception of voice individuality and physical correlates," IEICE Technical report, N85-03-9.

[Garofolo, 1988] Garofolo, J. S, "Getting Start with the DARPA TIMIT CD-ROM,".

[Griffin, 1984] D.W.Griffin, J.S.Lim, "Signal Estimation from Modified Short-Time Fourier Transform," IEEE ASSP Vol. ASSP-32,No.2.

[Hakoda, 1987] K. Hakoda, "Method of converting voice characteristics between male and female by modifying pole parameters," Proc. Fall Meeting of Acoust. Soc. Jap., 2-6-13, in Japanese.

[Hayashi, 1985] C. Hayashi, "Recent theoretical and methodological developments in multidimensional scaling and its related method in Japan," Behaviormetrika No. 18, 1095.

[Imai, 1980] S.Imai, "Log Magnitude Approximation(LMA) Filter, " IEICE Vol.J63-A No.12,1980, in Japanese

[Itoh, 1982] K. Itoh, S. Saito, "Effects of acoustical feature parameters of speech on perceptual identification of speaker," IEICE Vol. J65-A No.1, 101-108, in Japanese.

[Jelinek, 1980] Jelinek, F., et al., "Interpolated Estimation of Markov Source Parameters from Sparse data," Pattern Recognition in Practice, pp.381-397, E.S.Gelsema and L.N.Kanal, ed., North-Holland Publishing Company(1980)

[Juang, 1982] B-H. Juang, A. H. Gray, Jr., "Multiple stage vector quantization for speech coding," ICASSP82, pp. 597-600, May 1982.

BIBLIOGRAPHY

[Kawabata, 1990] Kawabata, T., et al., "Japanese Phonetic Typewriter Using HMM Phone Units and Syllable Trigram," ICSLP90, pp.717-720..

[Klatt, 1990] D. H. Klatt, L. C. Klatt, "Analysis synthesis, and perception of voice quality variations among female and male talkkers," J. Acoust. Soc. Am. 87(2), Feburary 1990, 820-857.

[Kullback, 1970] Kullback, S. "Information theory and Statistics", Jhon Wiley & Sons and Chapman & Hall.

[Kurematsu, 1987] Kurematsu, H. "Automatic Telephone Interpretation: A Basic Study", ATR Technical Report.

[Kuwabara, 1987] H. Kuwabara, T. Takagi, "Quality Control of Speech by Modifying Formant Frequencies and Bandwidth," 11th Inter. Congress of Phonetic Science, 281-284, 1987.

[Kuwabara, 1989] Kuwabara, H, Takeda K, Sagisaka Y, Katagiri S, Morikawa S, Watanabe T. "Construction of a large Japanese speech database and its management system", ICASSP, 560-563.

[Leung, 1985] Leung, C L, Zue W V. "Automatic Alignment of Phonetic Transcriptions with Continuous Speech" ,IASTED International Symposium on Robotics and Automation, 24-27.

[Levinson, 1985] S. E. Levinson, "Structual methods in automatic speech recognition," Proceedings of the IEEE, Vol. 73, No. 11, pp. 1625-1650, Nov. 1985.

[Linde, 1980] Linde, Y., Buzo A., Gray R. M., "An Algorithm for Vector Quantizer Design," IEEE Trans. on Communication,Vol.COM-28, pp.84-95, January.

[Lim, 1988] J. S. Lim, A. V. Oppenheim, "Advanced Topics in Signal Processing," Prentice Hall, Englewood Cliffs, New Jersey.

[Makhoul, 1985] J. Makhoul, S. Roucos, H. Gish, "Vector quantization in speech coding," Proceedings of the IEEE, Vol. 73, No. 11, pp. 1551-1588, Nov. 1985.

BIBLIOGRAPHY

[Matsuda, 1966] R.Matusda, "Effect of time vatiation of input signals on discrimination threshold of the transmission path with valley-shaped characteristics," Jour., I.E.C.E., Japan, 49, 10, pp. 1865-1871, Oct. 1966.

[Moriya, 1982] T. Moriya, M. Honda, "Vector quantization of the LPC residual signal in frequency domain," Trans. Committee on Speech Res. ASJ, S82-47, pp. 369-376, Nov. 1982.

[Nakajima, 1988] Nakajima, S., et al, "Automatic generation of synthesis units based on context oriented clustering," ICASSP-88, pp.659-662.

[Peterson, 1990] Peterson, P., et al., "Improving intelligibility of a 300 B/S segment vocoder," ICASSP-90, pp.653-656.

[Portnoff, 1981] M.R.Portnoff,"Time-Scale Modification of Speech Based on Short-Time Fourier Analysis," IEEE ASSP Vol. ASSP-29, No3.

[Price, 1989] P. Price, "Male and female voice source characteristics: Inverse filtering results", Speech comm., 8, 261-277, 1989.

[Rabiner, 1983] L. Rabiner, S. E. Levinson, M. M. Shondi, "On the application of vector quantization and hidden Markov models to speaker-independent isolated word recognition," Bell Syst. Tech, J., Vol. 62, pp. 1075-1105, Apr. 1983.

[Rosenberg, 1976] A. E. Rosenberg, "Evaluation of an automatic speaker - verification system over telephone lines", Bell Syst. Tech. J., 723-744, .

[Roucus, 1982] S. Roucos, R. Schwartz, J. Makhoul, "Segment quantization for very-low-rate speech coding," ICASSP82, pp. 1565-1569, May 1982.

[Roucos, 1985] S.Roucos, A.M.Wilgus, "High Quality Time-Scale Modification for Speech," ICASSP'85

[Ruspini, 1970] E. Ruspini, "Nemerical method for fuzzy clustering," inf., Sci., Vol.2, 1970.

BIBLIOGRAPHY

[Sagisaka, 1988] Sagisaka, S., "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," ICASSP-88, pp.679-682.

[Sato, 1974] H. Sato, "Acoustic cues of female voice quality," IEICE Vol. 57-A No.1, 23-30, in Japanese.

[Seneff, 1982] S.Seneff,"System to Independently Modify Excitation and/or Spectrum of Speech Waveform Without Explicit Pitch Extraction," IEEE ASSP Vol. ASSP-30, No.4.

[Shikano, 1986] K. Shikano, K. Lee, R. Reddy, "Speaker adaptation through vector quantization", ICASSP86, 2642-2646.

[Shiraki, 1988] Shiraki, Y., et al, "LPC speech coding based on variable-length segment quantization," IEEE ASSP Trans., Sept., 1988, pp.1437-1444.

[Soong, 1988] F. K. Soong, A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," IEEE ASSP-36, 6, 871-879.

[Soong, 1989] Soong, F. K., "A phonetically labeled acoustic segment(PLAS) approach to speech analysis-synthesis," ICASSP-89, pp.584-587.

[Sugiyama, 1981] M. Sugiyama, K. Shikano, "LPC peak weighted spectral matching measures," Electronics and Communications in Japan, Vol. 64-A, No. 5, pp. 50-58, 1981.

[Takagi, 1987] T. Takagi, T. Umeda, "A voice-quality conversion system by Fast Fourier Transform," Proc. Fall Meeting of Acoust. Soc. Jap., 2-6-12, in Japanese.

[Umeda, 1957] Umeda, N. NTT Electrical Communication Labs. Technical Report,No.579.

[Wong 1983] D. Y. Wong, B. H. Juang, D. Y. Cheng, "Very low data rate speech compression with LPC vector and matrix quantization," ICASSP83, pp. 65-68, Apr. 1983.

BIBLIOGRAPHY

[Zue, 1985] Zue, W V "Speech spectrogram reading", MIT special summer course lecture note.