Voice Transformation Using Principle Component Analysis Based LSF Quantization and Dynamic Programming Approach

Özgül Salor, Mübeccel Demirekler

Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

(salor, demirek}@metu.edu.tr

Abstract

The goal of voice transformation (VT) is to modify the speech of a source speaker such that it is perceived as if spoken by a target speaker. In this paper, we present a speaker specific line spectral frequency (LSF) quantization based on principle component analysis (PCA) and k-means clustering for VT. An LPC based source-filter model is used to model the speech. Transformation is applied to the spectral characteristics of the speaker, while pitch scaling is applied on the residual signal. PCA has been used to determine the principle components of the source and target LSFs to obtain a more efficient quantization. Only the dimensions with high variance have been quantized and those dimensions have been used to obtain the histogram matrix mapping the two speakers during training. To select the best target codeword sequence corresponding to a source codeword sequence in a sentence, a dynamic programming approach is used. Dynamic programming approach approximates the long-term behavior of LSFs of the target speaker, while it is trying to preserve the relationship between the subsequent frames of the source LSFs. Objective and subjective evaluations have shown that dimension reduction of LSFs before quantization and dynamic programming improves the voice transformation performance.

1. Introduction

The aim of voice conversion (VT) is to modify a source speaker’s speech such that it is perceived as if spoken by a target speaker. VT has numerous applications such as fast personification of text-to-speech systems and movie dubbing. The approach to the problem consists of a training phase where training speech data from the source and target speakers are used to formulate a spectral transformation that maps the acoustic space of the source speaker to that of the target speaker. Various solutions have been proposed to solve this problem [1,2,3,4]. In this paper, we propose a dimension reduction based on principle component analysis (PCA) and k-means clustering to obtain speaker-specific LSF codebooks for VT training. Only the dimensions with high variance have been quantized and those dimensions have been used to obtain the histogram matrix mapping the source speaker to the target speaker. To solve the problem of LSF discontinuities due to independent transformation of frames, a dynamic programming approach is used [5].

An LPC based source-filter model is used to model the speech. Transformation is applied on the spectral characteristics of the speaker, while pitch scaling is applied on the residual signal. A simple block diagram of the transformation system is given in Figure 1.

This paper is organized as follows: Section 2 presents the proposed method for LSF quantization. Section 3 explains the mapping histogram matrix preparation during training. Section 4 explains the transformation method and the dynamic programming approach. Finally, Section 5 presents the objective and subjective evaluation results on the proposed VT system.

2. PCA based LSF quantization

Quantization of LSFs has been achieved by k-means quantization algorithm [6] after a PCA based dimension reduction [7] applied to LSFs. LPC analysis has been applied to 180-sample non-overlapping speech frames sampled at 8 kHz. The database has two speakers of Turkish uttering 235 triphone-balanced sentences [8]. There are more than 30,000 frames of speech of each speaker after the removal of silence frames. A 10th order LPC analysis is performed on both the target and source speech signal using Hamming windows of length 200 samples centered on the last sample in the current frame. LSFs have been obtained from LPC coefficients at every frame. Source and target feature stream lengths have been equated using a dynamic time warping procedure, which selectively deletes or repeats frames from the target speaker feature stream to match the number of source frames within phonetically equivalent regions. PCA has been applied to voiced and unvoiced frames separately. Those matched LSF data sets are defined as $X^V$ and $X^U$ for the source, and similarly $Y^V$ and $Y^U$ for the target speaker. Superscripts $V$ and $UV$ stand for voiced and unvoiced respectively. $X^V$ and $Y^V$ are $N_V\times 10$ matrices with LSFs on the rows, where $N_V$ represents the number of matched voiced frames of the source and target speakers. Similarly, $X^U$ and $Y^U$ are of size $N_UV$, where $N_UV$ is the number of unvoiced frames.

PCA is applied on the four data matrices, $X^V$, $X^U$, $Y^V$, and $Y^U$. The aim of applying PCA is to obtain the principle components of the data and obtain projection matrices, $T_X^V$, $T_X^U$, $T_Y^V$, and $T_Y^U$, for voiced and unvoiced data matrices of source and target speakers respectively. Transformation matrices are of size $10 \times 10$ and their columns are the eigenvectors of the covariance matrices obtained from the zero-mean data as explained in [7]. Mean vectors of the data matrices $\mu^V$, $\mu^U$, $\mu^V$, and $\mu^U$ are represented with


\[ \mathbf{x}^V, \mathbf{x}^{UV}, \mathbf{y}^V, \text{ and } \mathbf{y}^{UV} \text{, respectively. Then the projections of the matrices } \mathbf{X}^V \text{ and } \mathbf{Y}^V \text{ are given as:} \]

\[ \tilde{\mathbf{X}}^V = \left( \mathbf{X}^V - \left[ \begin{array}{c} 1 \\ \vdots \\ 1 \\ \mathbf{x}^V \end{array} \right] \right) \mathbf{T}^V_x, \quad \tilde{\mathbf{Y}}^V = \left( \mathbf{Y}^V - \left[ \begin{array}{c} 1 \\ \vdots \\ 1 \\ \mathbf{y}^V \end{array} \right] \right) \mathbf{T}^V_y. \]

Similarly, \( \mathbf{X}^{UV} \) and \( \mathbf{Y}^{UV} \) are also projected to \( \tilde{\mathbf{X}}^{UV} \) and \( \tilde{\mathbf{Y}}^{UV} \).

The aim of using PCA here is to quantize the projected data more efficiently. In Figure 2, the first two dimensions of the mean-subtracted LSF values of voiced data \( (\mathbf{X}^V \text{ on the left and } \mathbf{Y}^V \text{ on the right}) \) are plotted with respect to each other on the upper panel. The lower plots are the first two components of \( \tilde{\mathbf{X}}^V \) and \( \tilde{\mathbf{Y}}^V \) (after the projection). Data dimensions are less correlated after projection and quantization will be much more efficient due to the increased diversity of data on the plane as observed in Figure 2.

![Figure 2: Plots of the first two components of the normalized LSFs in the upper panel and the first two components of the projected LSF components with respect to each other below.](image)

<table>
<thead>
<tr>
<th>Table 1: Eigenvalues obtained after PCA.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_{ix} )</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>3.48</td>
</tr>
<tr>
<td>1.66</td>
</tr>
<tr>
<td>1.22</td>
</tr>
<tr>
<td>0.87</td>
</tr>
<tr>
<td>0.49</td>
</tr>
</tbody>
</table>

Another use of PCA is that the number of dimensions to be quantized can be reduced after projection, which may result in more efficient quantization. First five eigenvalues obtained from PCA, \( e_{ix}^V, e_{iy}^V, e_{ix}^{UV} \), and \( e_{iy}^{UV} \) are given in Table 1. In PCA, if the eigenvector corresponding to an eigenvalue is high, then the data has a high variance along the dimension of that eigenvector. As observed in the table, the first few dimensions are very significant, while the others are less significant. Then k-means clustering algorithm given in [6] has been applied to the selected number of dimensions of the projected data to obtain a limited set of vectors representing the projected LSF space of the speakers individually. Let us define \( N \) as the number of principle components to be quantized and \( L \) as the number of codewords in k-means algorithm. Determination of the initial codebook in k-means clustering is achieved such that the data distribution in the projected LSF plane is reflected properly. Note that, to start with a good approximation of the data distribution for the initial codebook is preferred so that the algorithm converges to an optimum codebook fast. K-means iteration is applied to modify the initial codebook until the fractional drop in the average distortion becomes equal to or less than \( 2.22 \times 10^{-16} \) and new codebooks \( C_{ix}^V, C_{iy}^V, C_{ix}^{UV} \), and \( C_{iy}^{UV} \) are obtained.

3. Training

230 sentences have been used for training. Once the speaker-specific codebooks for voiced and unvoiced data of both speakers are determined, mapping histogram matrices are obtained. In the histogram matrix, the cell at \( i^{th} \) row and \( j^{th} \) column shows how many times \( i^{th} \) source codeword corresponded to \( j^{th} \) codeword of the target speaker in the database. Zero-mean and projected data form source and target speakers, \( \tilde{\mathbf{X}}^V, \tilde{\mathbf{X}}^{UV}, \tilde{\mathbf{Y}}^V, \text{ and } \tilde{\mathbf{Y}}^{UV} \) are quantized using codebooks \( C_{ix}^V, C_{iy}^V, C_{ix}^{UV} \), and \( C_{iy}^{UV} \), respectively. Mapping matrices are of size \( L \times L \). One example mapping histogram matrix obtained after quantization with \( N=4 \) (quantized principle components) and \( L=64 \) (number of codewords) for voiced speech data is presented in Figure 3.

![Figure 3: Mapping histogram matrix for quantized voiced frames of the source and the target speakers, with N=4, L=64.](image)

4. Transformation

In the transformation mode, the system makes a voiced-unvoiced decision for each frame. Then the mean LSF vector is subtracted from the LSF vector of that frame. Zero-mean LSF vector is projected and \( \tilde{\mathbf{X}}^V \) or \( \tilde{\mathbf{X}}^{UV} \) is obtained. \( N \) selected principle components (\( N \) can be from 1 to 10), are quantized using source's codebook. Note that quantization codebook is different for different \( N \) values, since codeword vectors are of size \( N \times L \). Those codebooks are obtained in the training mode. Each quantized frame is mapped to a target codeword based on the mapping histogram matrix. Dynamic programming, which is described in the next section, is applied along sentences to map the frames of the source to those of the target using the mapping histogram matrix, \( H \). Dynamic programming has been shown to improve the transformation performance significantly compared to a baseline system, which maps the maximum of the mapping histogram matrix [5,9].

4.1. Dynamic programming approach

Dynamic programming helps to use the histogram matrix, \( H \), obtained during training, more efficiently during transformation. Moreover, it lets the transformed LSF values follow the LSF continuities or discontinuities between the subsequent frames of the source speaker, which increases the synthetic speech quality.
Codewords of the source and target speakers are shown by $C_s = \{x_1, x_2, \ldots, x_N\}$ and $C_t = \{y_1, y_2, \ldots, y_M\}$ respectively. The first step is to quantize the source speaker’s frames along a sentence to obtain the $x[n]$ vector, whose elements are the indices of the codewords in $C_s$. $n$ shows the frame index in the sentence. Then the sentence histogram matrix, $H^{sen}$, is obtained as shown below:

$$
H^{sen} = 
\begin{bmatrix}
H(x[1], 1) & H(x[2], 1) & \cdots & H(x[M], 1) \\
H(x[1], 2) & H(x[2], 2) & \cdots & H(x[M], 2) \\
\vdots & \vdots & \ddots & \vdots \\
H(x[1], L) & H(x[2], L) & \cdots & H(x[M], L)
\end{bmatrix},
$$

where $M$ shows the total number of frames in the sentence. Dynamic programming finds the highest probability path from $n=1$ to $n=M$ on the above matrix under the constraint, which is the LSF distance between the subsequent frames of the source speaker. The parameters used in determining the best path are the transition probabilities of the target speaker from one codeword to the other, $T(i,j)$, and the normalized sentence histogram matrix $H^{sen}$. The probability of transition from target codeword $y_i$ to $y_j$ is shown by $T(i,j)$ and this matrix is obtained from the corpus of the target speaker during training. The columns of $H^{sen}$ matrix is normalized to add up to unity, so that the columns show the LSF probabilities of the target LSF codewords corresponding to the source LSF $x[n]$. This normalized matrix is called $P$. The method is illustrated with an example in Figure 4. To determine the best path towards the node, for example, $P(L,2)$, first all the allowable paths towards $P(L,2)$ are determined.

![Dynamic programming example](image)

Allowable paths are the paths which satisfy the constraint given below as:

$$SD(x[1], x[2]) − D ≤ SD(y_j, y_{j+1}) ≤ SD(x[1], x[2]) + D,$$

where $D$ is the allowable distance among subsequent frames. $SD(x, y)$ is the Bark-weighted RMS error in dB between the power spectra of $x$ and $y$. This constraint lets the target LSFs follow a smooth path from one frame to another, when the source LSFs are changing smoothly from frame to frame. At the same time, it forces the target LSFs to follow the discontinuities between the subsequent frames of the source speaker (for example, between frames of plosive phonemes). Once the allowable paths are determined, the path probability from $y_j$ to node $P(L,2)$, which is defined as $P_{path}(L,2)$, is obtained as $P_{path}(L,2) = \sum_{n=1}^{M} P_{path}(j,n) \cdot T(j,l) \cdot P(L,2)$, assuming $y_j$ is among the allowable paths. $P_{path}(j,n)$ is the probability of being at point $j$ at frame $n$. Path towards $P(L,2)$ is selected among $y_j$s which give the highest $P_{path}$ value. The final column of the $P_{path}$ matrix obtained at the end of the sentence has the accumulated probabilities along the sentence. The highest probability row at the final column of $P_{path}$ is selected and the path from $n=M$ to $n=1$ is selected that point gives the sequence of transformed LSF codewords.

4.2. Formulation of the Transformation

Voiced and unvoiced superscripts will be dropped for the sake of simplicity in this section. Let us define the $10x10$ projection matrix obtained after PCA from the source data as $T_s = [T_s^1, T_s^2]^T$, where $T_s^1$ is the $10xN$ matrix, which transforms the first N principle components of the zero-mean LSF vector of the source. Then the target codeword obtained after transformation is given as $f(Q_s^c(x−\bar{T}_s^c))$, where $f(\cdot)$ denotes the mapping operation using dynamic programming, and $Q_s^c$ denotes the quantization operation of source using $N$ principle components of the projected LSFs. $\bar{T}_s$ is the mean LSF vector of the source. However, this vector is an $Ns$ vector selected from the target codewbook, $C_t^c$, so it should be back projected with the inverse transformation matrix of the target for the first $N$ components, $T_s^1$. Note that $(T_s^1)^T(T_s^1)^{-1} = \bar{T}_s^c$ since columns of $T_s^1$ and $T_s^2$ are unit eigenvectors. Then the estimated target LSF vector $y_{est}$ is given as $y_{est} = [f(Q_s^c((x−\bar{T}_s^c)))](T_s^1)^T + \bar{y}$, where $\bar{y}$ is the mean vector of the LSFs of the target.

4.3. Speech Synthesis

Synthesis is achieved by replacing the source LSF codewords with the transformed LSFs at every frame during synthesis. Pitch modification is applied in the residual signal to match the pitch range of the target speaker. Using the maximum and the minimum values of the two speakers in the 230-sentence corpus, a linear relationship between their pitch periods have been obtained. Then synthesis is achieved by driving the transformed LPC filter with the pitch-modified residual.

5. Results and Conclusions

5.1. Objective Evaluations

The VT performance measure used to evaluate our VT system is based on the comparison of the spectral distance between the source and the target speakers, $SD(s,t)$ with the spectral distance between the converted speech and the target speaker, $SD(c,t)$. Evaluations have been done on a test set of ten sentences, which are not in the training set of 230 sentences. The performance index, $PI$, used is given as:

$$PI = 1 - \frac{E(c,t)}{E(s,t)} = 1 - \frac{1}{M} \sum_{m=1}^{M} SD(LSF_m^s, LSF_m^c).$$

where $E(c,t)$ is the average spectral distance between LSF vectors of converted and target, over all $M$ frames, and similarly $E(s,t)$ is the average spectral distance between LSF vectors of source and target. The performance index $PI$ is unity in case of perfect transformation, and approaches towards zero as the performance of the system degrades. A PI smaller than zero shows an unsuccessful transformation.
There exist two training parameters: number of principle components to be quantized, \(N\), and the number of codewords, \(L\), which is also the size of the mapping histogram matrix. We varied \(N\) = 4, 6, 8, 10 for voiced and unvoiced data and \(L=64, 128, 256\) for voiced data and \(L=16\) for unvoiced data. Mapping histograms and codewords have been determined for both speakers for each case. There is one transformation parameter, \(D\), which is the interval to obtain the allowable paths in the dynamic programming algorithm given in Section 4.1. \(D\) has been varied from a finite value towards infinity. \(D\) equals infinity means that all paths are allowed in the dynamic programming procedure.

\[ \text{Figure 5: Performance indices for transformation from Speaker-1 to Speaker-2, with } L=64 \text{ above and } L=96 \text{ below.} \]

5 sentences not included in the train set have been used for testing. Figure 5 illustrates the LSF performance indices, \(PI\), for different codebook sizes (\(L\)) and the number of principle component used for quantization (\(N\)). It has been observed that performance indices are higher than those obtained in our previous work based on using MELP’s LSF codebook to obtain speaker-specific codebooks given in [5] and [9]. Highest performance index in [5] is 0.26, while it is 0.36 with \(L=64\) and \(N=4\), and \(D=75\) in this work.

Test results present that both reducing the number of principle components to be quantized and using dynamic programming during transformation increase transformation performance. Using LSF dimensions with high variance after projection results in a better mapping between the two speakers, although the effect of the less significant dimensions are neglected. It is observed that decreasing the number of codewords below 64 reduces the transformation quality, since it causes a coarse representation of the LSF spaces of the speakers. Increasing the number of codewords, on the other hand, requires the number of data to be increased to obtain a successful mapping.

In Figure 5, it is also observed that increasing \(D\) reduces \(PI\). Increasing \(D\) approaches to the case where no constraints are used in the dynamic programming. This shows that the use of dynamic programming improves the performance.

5.2. Subjective Evaluations

An ABX test has been used to evaluate the speaker-similarity performance of the VT system. In the ABX test, \(A\) and \(B\) represent the original speakers, and \(X\) is the transformed speech either from \(A\) to \(B\) or from \(B\) to \(A\). Subjects are asked to determine whether \(X\) is more similar to \(A\) or \(B\). The test includes 6 transformed sentences and 3 transformations each for \(L=64\) and 96. These are the \(L\) values which result in the highest performance indices, \(D\) is 75, which gives the highest performance indices for most of the cases. For 3 of the transformed sentences \(N\) has been equated to 4, and for the other 3 sentences \(N\) is 10. This is to observe the difference in perceptual results after dimension reduction. 2 of the transformations are from Speaker-2 to Speaker-1 and the remaining 4 transformations are from Speaker-1 to Speaker-2. 20 subjects have taken the test. 118 converted sentences have been detected as the target speaker out of 120 sentences in total. 2 of the incorrect decisions were for \(L=96\) and \(N=10\). Since no errors have been detected with \(N=4\) case, it is possible to say that dimension reduction increases the system performance. Correct speaker detection rate is 98.3%.

6. Acknowledgements

Authors thank TÜBİTAK, Scientific and Technical Research Council of Turkey, for supporting this work through a combined doctoral scholarship.

7. References