A DYNAMIC PROGRAMMING APPROACH TO CONTEXT-FREE VOICE TRANSFORMATION

Özgül Salor, Mübeccel Demirekler

Department of Electrical and Electronics Engineering, Middle East Technical University, Turkey
phone: + (90) 312-2104419, fax: + (90) 312-2101261, email: {salor,demirek}@metu.edu.tr
web: www.eee.metu.edu.tr

ABSTRACT

In this paper, we present a dynamic programming approach to voice transformation (VT). The goal of VT is to modify the speech of a source speaker such that it is perceived as if spoken by a target speaker. The speech model used in this work is based on MELP (Mixed Excitation Linear Prediction) speech coding algorithm. The designed system obtains speaker-specific codebooks of line spectral frequencies (LSFs) out of MELP's multi-stage vector quantization LSF codebook for both source and target speakers. Those codebooks are used to train a mapping histogram, which is used for LSF transformation from one speaker to the other. The baseline system uses the maxima of the histograms for LSF transformations. The shortcomings of this system, which are the limitation of the target LSF space and the spectral discontinuities due to independent mapping of subsequent frames, have been overcome by applying the dynamic programming approach. Dynamic programming approach tries to model the long-term behaviour of the LSFs of the target speaker, while it is trying to preserve the relationship between the subsequent frames of the source LSFs, during transformation. Both objective and subjective evaluations have been conducted and it has been shown that dynamic programming approach improves the performance of the system in terms of both the speech quality and speaker similarity.

1. INTRODUCTION

In this paper, we propose a dynamic programming approach for voice transformation (VT). The aim of VT is to modify a source speaker's speech such that it is perceived as if spoken by a target speaker. The proposed dynamic programming approach tries to preserve the long-term behaviour of the line spectral frequencies (LSFs) of the target speaker, while it is considering the distance between the LSFs from subsequent frames of the source speaker during transformation. MELP (mixed excitation linear prediction) speech coding algorithm has been used as an analysis and synthesis framework [1, 2]. Using MELP's multi-stage vector quantization (MSVQ) codebook, we have obtained speaker-specific LSF codebooks for source and target speakers. In addition to our previous work on VT in [3], MELP's MSVQ LSF codebook has been reduced to speaker-specific codebooks and the dynamic programming approach has been introduced. Section 2 presents the method for obtaining the speaker-specific codebooks. Section 3 explains the dynamic programming approach and finally Section 4 presents the objective and subjective evaluation results on the proposed VT system.

2. OBTAINING THE NEW CODEBOOKS

We have considered modifying the spectral characteristics of the source speaker for VT. In MELP, a 10th order linear prediction analysis is performed on the input speech signal using a 200-sample (25 ms) Hamming window centered on the last sample in the current frame. A MELP frame interval is 22.5 ms in duration and 180 voice samples (for 8 kHz sampling rate). MELP uses 4-stage vector quantized LSFs to code the Linear Prediction Coefficients (LPCs). The quantized LSFs for each frame is obtained by summing the frequencies selected from each stage of the LSF codebook. The first stage consists of 128 LSF vectors, while other 3 stages have 64 frequency vectors each. The first stage LSF vectors form the general shape of the LPC spectrum, while the frequency vectors in the other stages are added to the first to get a more detailed spectral shape.

2.1 Speech Corpus for VT

For training, speech collected from two male speakers of Turkish, has been used. 235 sentences from a triphone-balanced sentence set [4] uttered by both speakers have been recorded. Phoneme level alignments have been provided using the *Sonic Turkish Aligner* [3]. The phoneme-level alignments have been used to time-align the MELP-frames of the two speakers. Dynamic time-warping (DTW) has been used to equate the number of the source frames and that of the target frames. DTW achieves this procedure by selectively deleting or repeating frames from the target speaker feature stream to match the number of source frames within phonetically equivalent regions defined by phoneme-level alignments. Note that phoneme-level alignments have been used only for DTW procedure here. The training and transformation parts of the proposed VT system are context-free. The number of time-aligned frame pairs of the source and target speakers is approximately 30,000.

Cross occurrence histograms of all stages of MSVQ indices from the VT corpus have been obtained. Figure 1 illustrates the histogram matrix mapping the first stage indices of the two speakers. x-axis shows the 128 first stage indices of Speaker-1 while y-axis shows those of Speaker-2. z-axis shows the occurrence numbers in the 30,000-frame VT corpus.

2.2 Observations

MSVQ indices of the LSFs for every time-aligned MELP-frame of both the source and the target speakers are extracted for analysis. Assuming each stage *i* of MSVQ is a random variable, X_i and Y_i for Speaker-1 and Speaker-2 respectively, mutual information analysis has been done to investigate the amount of information one speaker's LSF indices give about those of the other. Let us define the empirical probabilities of the LSF indices of Speaker-1 and Speaker-2 as $p_i(x)$ and $p_i(y)$ respectively, where the subscript *i* corresponds to the stage numbers of MSVQ. Joint distribution $p_i(x,y)$ is obtained from the mapping histogram of the corresponding stage (mapping histogram of the first stage is observed in Figure 1) by normalizing it with the total frame number.

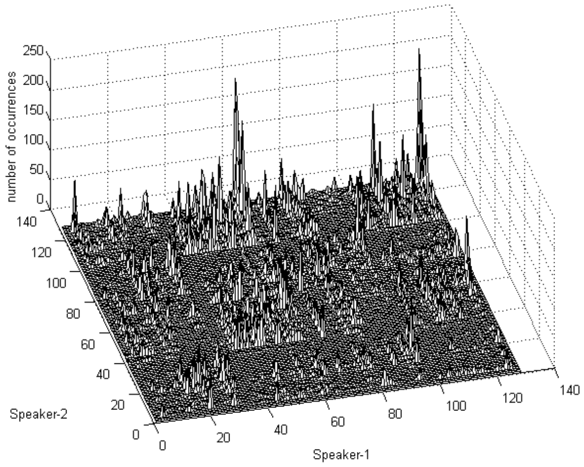

Figure 1. Histogram matrix mapping the first stage indices of MELP's multi-stage LSF quantization of the Speaket-1 and Speaker-2.

Mutual information, *$I(X_i,Y_i)$*, is computed as:

$$I(X_i,Y_i)=\sum_x\sum_y p_i(x,y)\log_2\frac{p_i(x,y)}{p_i(x)p_i(y)} \qquad (1).$$

$I(X_1,Y_1)$ has been found to be 1.7 bits, while *$I(X_2,Y_2)$* is 0.21. This shows that mapping the MSVQ stages of one speaker to those of the other independently, will not result in a successful transformation, because second MSVQ stage of one speaker does not give enough information about the second stage of the other speaker. When all combinations of the first two stages are considered together (i.e. 128x64=8192 LSF indices), the mutual information between the combined stages of the two speakers is obtained as 2.23 bits. This shows that mapping the two stages dependently results in a more successful VT in terms of LSF mapping. However, considering all four stages together is not feasible in terms of our corpus size. All four LSF stages of MELP yield approximately 33 million possibilities, while we have 30,000 frames in our VT corpus.

Another observation is that some LSF indices are not used by some speakers. This can be observed in Figure 1. Some columns and rows are completely empty. The unused codewords cause inefficiency of codebook usage while mapping the LSF space of one speaker to the other. Therefore, we have considered obtaining speaker-specific LSF codewords out of MELP's LSF codewords.

2.3 Speaker-Specific LSF Codeword Selection

Considering the observations presented in the previous section, we have decided to obtain speaker-specific codebooks with reduced number of codewords out of MELP's 4 stage LSF quantizer. The method used can be summarized in the following steps:

- The first two stages provide 8192 different LSF vectors. 1600 of them, specific to the speaker, are enough to cover 80% of the LPC spectrum space of each speaker. The most frequently used 1600 two-stage combinations for each speaker are obtained.
- New 3rd and 4th stage indices for the whole corpus are determined once more, forcing MELP to use the selected 1600 1st and 2nd stage combinations.
- Considering the 3rd stage with those 1600 two-stage indices makes 1600x64=102400 LSF combinations. 102400 LSF vectors have been reduced to *L*, by choosing the most frequently used L combinations for each speaker. The rest (102400-*L* combinations) are mapped to one of those *L* LSF vectors. 4th stages are neglected in this quantization, since 4th stage frequencies of LSF has the least effect on the final shape of the LPC spectrum. During transformation, the 4th stage frequencies of the source speaker are added to the transformed 3-stage LSF vector (which is one of the *L* codewords of the target speaker) directly.

Once the *L* codewords are obtained for both speakers, the VT corpus is quantized once more using the new codewords for each speaker. The *LxL* histogram matrix mapping the new LSF indices of the two speakers has been obtained.

2.4 Transformation and the Baseline System

In the baseline system, during the analysis phase, analysis every frame of the source speaker is quantized to one of *L* codewords of the source. During LSF transformation, every codeword of source speaker is mapped to the target codeword which has the highest occurrence rate in the histogram matrix corresponding to the source codeword. There are two shortcomings of this method: First one is the limitation of the LSF space of the target speaker to *L* codewords. The second one is the possibility of LSF discontinuities appearing at the frame boundaries due to mapping the subsequent frames independent of each other. Dynamic programming has been integrated to the baseline system to avoid these shortcomings.

Synthesis is achieved by replacing the source LSF codewords with the transformed LSFs at every frame during synthesis. After the spectral modification, pitch modification is applied in the residual signal to match the pitch range of the target speaker. Pitch modification is achieved in the MELP synthesis framework. 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. This relation has been used to modify the pitch value obtained by MELP analysis at every frame during synthesis.

3. 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 $\bar{x}=[x_1,x_2,\Lambda\ ,x_L]$ and $\bar{y}=[y_1,y_2,\Lambda\ ,y_L]$ 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 $\bar{x}$. *n* shows the frame number in the sentence.

Then the sentence histogram matrix, H^{sen}, is obtained as shown in Equation 2:

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

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. Probabilities are the empirical probabilities obtained from the occurrence rates of subsequent target LSF codewords. 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 2. To determine the best path towards the node, for example, $P(L,2)$, first all the allowable paths towards $P(L,2)$ are determined.

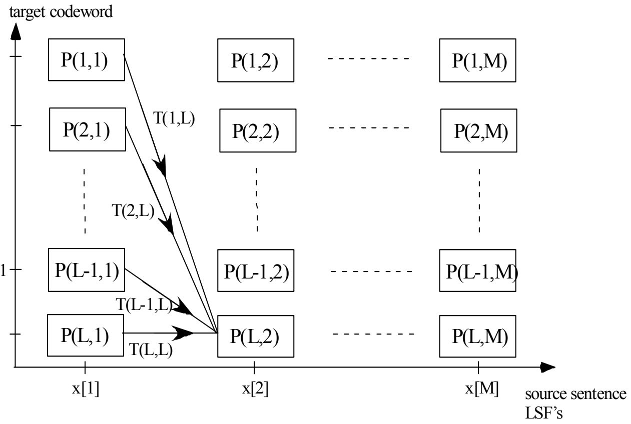

Figure 2. Dynamic programming example for LSF transformation.

Allowable paths are the paths which satisfy the constraint given below in (3).

$$SD(x[1],x[2]) - D \le SD(y_j, y_L) \le SD(x[1],x[2]) + D, \quad (3)$$

where D is the allowable amount of distance. $SD(x, y)$ is the spectral distance between the LSF vectors x and y, which is computed by using the spectral distance measure defined in MELP algorithm [1]. It is given in detail in Section 4 in Equation (7). 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 $P(L,2)$, which is defined as $P_{path}(L,2)$, is obtained as:

$$P_{path}(L,2) = P_{path}(j,1) \times T(j,L) \times P(L,2), \quad (4)$$

assuming y_j is among the allowable paths. $P_{path}(j,k)$ is the probability of being at point j at frame k. 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 $n=1$ to $n=M$ towards that point gives the sequence of transformed LSF codewords.

During transformation, the 4th stage frequencies of the source speaker are added to the 3-stage LSF vector (which is one of the L codewords of the target speaker) directly after transformation. This corresponds to moving the transformed LSF in the same direction with the 4th stage of the source speaker. The method can be illustrated in the two dimensional LSF space as seen in Figure 3. Block diagram of the whole transformation system is given in Figure 4.

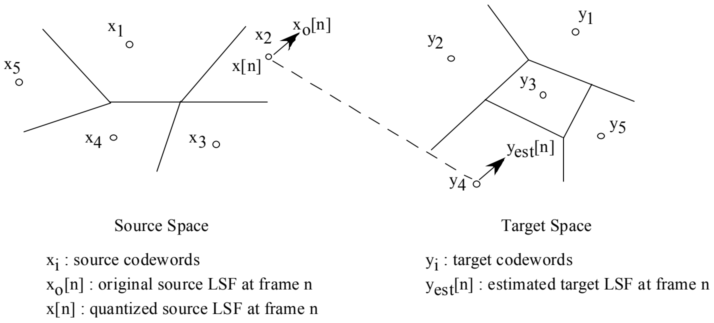

Figure 3. Illustration of the transformation in the two dimensional LSF space.

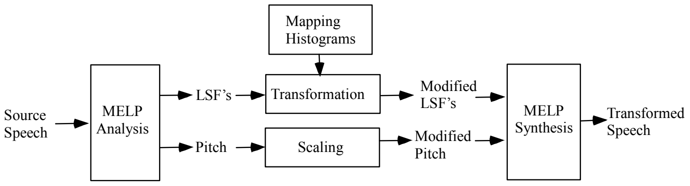

Figure 4. Block diagram of the VT system.

4. EVALUATION RESULTS AND CONCLUSIONS

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 five sentences, which are not in the training set of 230 sentences. The performance index used is given as:

$$P_{LSF} = 1 - \frac{E(c,t)}{E(s,t)}, \qquad (5)$$

$$E(x,y) = \frac{1}{M}\sum_{m=1}^{M} SD(LSF_x^m, LSF_y^m), \qquad (6)$$

where *E(x,y)* is the average spectral distance between LSF vectors of *x* and *y*, over all *M* frames. Spectral distance between 10th order LSFs f and $\hat{f}$ is computed as:

$$SD(f,\hat{f}) = \sqrt{\sum_{i-1}^{10} w_i (f_i - \hat{f}_i)^2}, \qquad (7)$$

where,

$$w_i = \begin{cases} P(f_i)^{0.3}, & 1 \le i \le 8 \\ 0.64P(f_i)^{0.3}, & i = 9 \\ 0.16P(f_i)^{0.3}, & i = 10 \end{cases}. \qquad (8)$$

This is the same perceptual distance used by MELP for quantizing LSFs [1]. *P(f$_i$)* is the inverse prediction filter power spectrum evaluated at frequency *f*$_i$.

The performance index P_{LSF} is 1 in case of perfect transformation, and approaches towards zero as the performance of the system degrades. A performance index smaller than zero shows an unsuccessful transformation.

We have two parameters in our VT system: Number of codewords in the codebook, *L*, and the allowable distance, *D*. *L* has been varied *L*=256, 128, 96, and 64. *D* has been varied *D*=0.14, 0.16, 0.128. 0.2, 0.4, 0.6, and ∞. *D*=∞ means using no constraints in the dynamic programming and it approximates the performance of the baseline system. The maximum value for *D* is obtained as 2.3 dB, which is the maximum distance between any two LSFs of the source speaker, and values *D*>0.6 result in the same performance indices, which is very close to the performance of the baseline system.

Figure 5 illustrates the performance indices obtained for *L*=256, 128, and 96 for VT from Speaker-1 to Speaker-2. *L*=64 results in performance indices smaller than zero. The reason is though to be the method used to obtain speaker-specific LSF codebooks from out of MELP's MSVQ LSF codebook. Since the reduction method is based on selecting the most frequently used LSF indices, when *L* is too small, selected codewords may be very close to each other, which may cause inefficient quantization of the LSF spaces of the speakers. The results with L=256 are worse than those with L=128. This is because more data is needed to approximate the correct occurrence probabilities as the number of codewords is increased. So there is a trade-off between selecting high and low number of codewords. Direct quantization on speech instead of using MELP's LSF codebook is considered as future work to improve the VT system performance. It is also observed in Figure 5 that dynamic programming improves the system performance.

Subjective listening tests have also been conducted using the same test sentence set. An ABX test has been done, where A and B are the source speakers' utterances, and X is the transformed speech. 20 subjects have taken the test. Subjects are allowed to listen to one original sentence from each of the two speakers as many times as they like until they get used to the speakers and they are also allowed to listen to them during the test. The original sentences are different than the transformed ones to prevent the long-term behaviour of the intonation of the speakers from effecting the decisions of the subjects. Then they listen to nine transformed sentences, and decide which speaker each one is. 3 groups of 3 sentences each with *L*= 256, 128 and 96 have been transformed either from Speaker-1 to Speaker-2 or vice versa. *D* values that result in the highest objective performance indices for each *L* have been used. It has been observed that out of 180 converted sentences tested, 174 have been perceived as the target speaker. 4 of the incorrect decisions were with *L*=128 and 2 of them were with *L*=96. We can conclude that the best perception has been obtained with *L*=128, which is also the case for objective evaluation results.

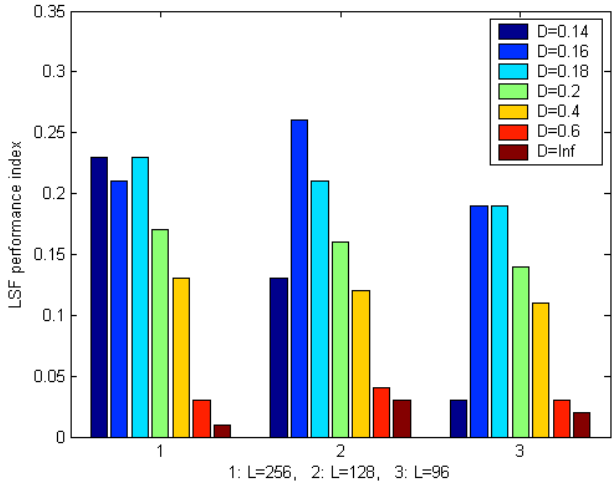

Figure 5. Performance indices of voice transformation from Speaker-1 to Speaker-2.

5. ACKNOWLEDGEMENTS

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

REFERENCES

[1] Federal Information Processing Standards Publication (MELP), Specifications for the Analog to Digital Conversion of Voice by 2400 Bit/Second Mixed Excitation Linear Prediction, Draft June 1997.

[2] McCree, A. V., Barnwell T., P., "A Mixed Excitation LPC Vocoder Model for Low Bit Rate Speech Coding", IEEE Trans. On Speech and Audio Processing, Vol. 3, No. 4, pp. 242-250, July 1995.

[3] Salor, Ö., Demirekler, M., "Spectral Modification for Context-Free Voice Conversion Using MELP Speech Coding Framework", International Symposium on Intelligent Multimedia, Video and Speech Processing, Oct. 2004.

[4] Salor, Ö., Pellom, B., Çiloğlu, T., et.al., "On Developing New Text and Audio Corpora and Speech Recognition Tools for the Turkish Language", in Proceedings of the International Conference on Spoken Language Processing, Denver, USA, Sep, 2002.