Transforming Voice Quality

Ben Gillett and Simon King

Centre for Speech Technology
University of Edinburgh, United Kingdom
beng@cstr.ed.ac.uk, Simon.King@ed.ac.uk

Abstract

Voice transformation is the process of transforming the characteristics of speech uttered by a source speaker, such that a listener would believe the speech was uttered by a target speaker. In this paper we address the problem of transforming voice quality. We do not attempt to transform prosody.

Our system has two main parts corresponding to the two components of the source-filter model of speech production. The first component transforms the spectral envelope as represented by a linear prediction model. The transformation is achieved using a Gaussian mixture model, which is trained on aligned speech from source and target speakers. The second part of the system predicts the spectral detail from the transformed linear prediction coefficients. A novel approach is proposed, which is based on a classifier and residual codebooks. On the basis of a number of performance metrics it outperforms existing systems.

1. Introduction

One of the main applications of voice conversion is in the field of text-to-speech adaptation. Modern speech synthesisers require a large database of speech. A voice transformation system which could be trained on relatively small amounts of data would allow new voices to be created with much lower cost. In addition, such a system could be used in a situation where the speaker was not available and previous recordings had to be used, such as is the case where a patient had lost the power of speech through disease or injury. Voice transformation also has other applications such as very low bandwidth speech encoding, multimedia entertainment, as a pre-processing step to speech recognition and also in the field of voice disguise. In addition, gaining a better understanding of the ways in which speakers differ is likely to be valuable more generally in both speech synthesis and recognition.

There has been a considerable amount of research directed at the problem of transforming voice quality [1, 2]. The general approach has been to begin with a training phase in which material from source and target speakers is aligned and used to define a transformation which maps the acoustic space of the source speaker to that of the target.

2. Speech data

Data from the Boston University Radio News corpus as described in [3] was used to both train and test the system. We performed endpointing on all waveforms. Four speakers, (two male and two female) were selected for the experiment. They are labelled as f1a, f2b, m1a, m2b within the corpus. The speakers are all native speakers of English and have North American accents. They are all professional news readers.

Experiments were run on the following transformation combinations; m1 to m2, m2 to m1, f1 to f2 and f2 to f1. T_{train} seconds of data were used for training. The test set consists of one minute of speech. The training and testing sets do not intersect. All performance measures presented here are for the test set.

3. Analysis

We carry out a pitch synchronous frame based analysis of the speech, since for short segments of speech the spectrum may be considered to be stationary. The speech was divided into short overlapping frames, where each frame was two pitch periods long and was centred around the current pitchmark. These frames were then windowed using a Hanning window. The Linear Prediction Coefficients (LPC) of the filter were computed using the autocorrelation method [4]. The order of LPC analysis, O_{LPC} was one of the variables of the experiment. The LPC filter coefficients were converted into line spectral frequencies (LSFs) [4]. Line spectral frequencies have better interpolation characteristics, which is important for this system since the target LSFs will be formed from a weighted sum of source LSFs.

The ear has better frequency resolution at lower frequencies [4]. In order that the numerical distance between a pair of LSFs better reflect the perceptual distance between them, this non-linear frequency resolution must be accounted for. One scale that achieves this is the Bark scale. The Bark warping function b is as follows:

$$b(f) = 6.log(\frac{f}{1200} + \sqrt{(\frac{f}{1200})^2 + 1})$$

The Bark-warping process was applied to the LSFs for each frame of speech. Residuals were computed by inverse filtering each frame of speech using the associated LPCs.

Time-alignment was carried out on each set of sentences for each source/target speaker pair. Firstly, Cepstral Coefficients (CCs) [4] for each Bark-warped set of LSFs were calculated, together with the log of the associated residual energy. The Dynamic Time-Warping (DTW) [4] algorithm was used to find the minimum error alignment.

4. Transforming the spectral envelope

As mentioned earlier, there are two components of our system, corresponding to the two components of the source-filter model. In this section we describe the component which transforms the spectral envelope (i.e. the filter).

4.1. Training

The purpose of the training stage is to estimate the parameters of a transformation function that will map source features (LSF

vectors) to target features with minimum error.

4.1.1. Fitting the GMM

Pre-GMM estimation rejection of poorly matched data
There is a great deal of variability within and across speakers as to the way words such as 'the' and 'a' are spoken. In some cases these words are drastically shortened, and in others they are even left out entirely. We attempt to reject poorly matched data, unlike all previous approaches. Two strategies were employed to help isolate the poorly matched frames.

Those pairs of aligned frames of speech where one speaker's speech was voiced and the other speaker's was unvoiced were rejected from the training set. If they have different voicing classification, this suggests that they were poorly aligned. As described in Gillett's thesis [5], the predicted amplitude envelope of the target is computed by modifying the source amplitude envelope. The frames of speech where the predicted amplitude is more than three times larger or smaller than the actual amplitude at that point are also rejected. When combined, these methods typically reject about 25% of the data, and were found to significantly improve quality in an informal listening test.

Estimation of the transformation function
The transformation function must map the features of the source speaker to the appropriate target speaker features. Gaussian mixture models are one possible approach to this problem. They have the useful property of being continuous, as opposed to a lookup table based approach such as that of Arslan and Talkin [1]. It has been shown that GMMs have as good as or superior performance at the task of voice transformation to other transformation approaches based on neural networks, vector quatization or linear regression [6].

We use the joint density approach as applied to VT by Kain [7]. This approach involves fitting a GMM to the joint density $P(x, y)$ and then predicting y from x by finding $E[y|x]$ (the expected value of y given x). To do this we form a vector Z where each element is composed of the source features X and target features Y, where $Z = \begin{bmatrix} X \\ Y \end{bmatrix}$

The probability distribution of a GMM with Q_{LSF} components is given by:

$$p_{GMM}(x; \alpha; \mu; \Sigma) = \sum_{q=1}^{Q_{LSF}} \alpha_q N(x; \mu_q; \Sigma_q), \sum_{q=1}^{Q_{LSF}} \alpha_q = 1, \alpha_q \geq 0$$

where α_q is the weight for component q, $N(x; \mu_q; \Sigma_q)$ is the n-dimensional normal distribution with mean μ_q and covariance Σ_q.

The probability of a datapoint x belonging to a particular class p may be computed using Bayes' rule, which is

$$P(c_p|x) = \frac{\alpha_p N(x; \mu_p; \Sigma_p)}{\sum_{q=1}^{Q_{LSF}} \alpha_q N(x; \mu_q; \Sigma_q)}$$

The Expectation Maximization (EM) algorithm is an iterative algorithm which may be used to find the most likely GMM parameters (α, μ, Σ) for a given set of data. To start the process we set α_q equal to $1/Q_{LSF}$ for all $q = 1...Q_{LSF}$, Σ_q equal to the identity matrix for all q, and set each μ_q by applying the K-means algorithm. The EM algorithm was then run until either the likelihood $P_{GMM}(x; \alpha; \mu; \Sigma)$ was maximized, or 30 iterations were exceeded.

Post-GMM estimation rejection of poorly matched data
Once the GMM had been fitted to the training data, a second stage of rejecting poorly matched data took place. We rejected $R\%$ of the data which had lowest probability $P(c_p|x)$ under the GMM. These points may be regarded as remaining outliers and are due to poor alignment. A GMM was then re-estimated for the remaining data points. The optimum proportion for rejection (15%) was found through informal listening tests. The appropriate amount to reject is likely to depend on the extent to which the source and target speakers' accents and prosody differ, since we wish to exclude those cases where the two speakers are saying very different things.

4.2. Transformation

In order to carry out transformation, the speech is first analyzed by computing Bark-warped LSFs for each frame. X and Y are the aligned source and target feature streams. There is a simple 1:1 mapping between X and Y. For each frame of source LSFs, the most likely target LSFs are computed. The expected value of the target LSFs for a target frame, y, may be computed using the appropriate source frame LSFs x as follows:

$$E[y|x] = \sum_{q=1}^{Q_{LSF}} (\mu_q^Y + \Sigma_q^{YX}(\Sigma_q^{XX})^{-1}(x) - \mu_q^X) P(c_q|x)$$

where

$$\Sigma_q = \begin{bmatrix} \Sigma_q^{XX} \Sigma_q^{XY} \\ \Sigma_q^{YX} \Sigma_q^{YY} \end{bmatrix} \quad \mu_q = \begin{bmatrix} \mu_q^X \\ \mu_q^Y \end{bmatrix}$$

After the predicted LSFs have been computed, a smoothing function is applied to each of the LSF coefficients, in order to restrict the difference in value between neighbouring frames. The filter used is a 2nd order lowpass digital Butterworth filter where the cutoff F_{LP} is expressed as a fraction of half the sampling rate. The sampling rate is the rate of pitchmarking. This low pass filtering of the LSFs is motivated by the fact that the components of the human speech system responsible for filtering the signal from the glottis are restricted in how rapidly they may change their response.

4.3. Synthesis

Once a vector of target LSFs has been predicted, the LSFs are then converted from Bark to Hertz and converted to LPCs. The associated target residuals are then found (see section 5), and a Hanning window is applied prior to inverse filtering with the associated LPC parameters. The resulting speech is then created by PSOLA (pitch synchronous overlap-add) of all the frames of windowed speech.

5. Transforming the spectral detail

In this section we address the problem of transforming the spectral detail which is represented by the residual. The system predicts residuals from transformed LSFs which were predicted in section 4. The source-filter model is based on an assumption that the residual is independent of the spectral envelope. However, we will show that the residual is sufficiently correlated with the spectral envelope that prediction is possible.

5.1. Training

The system only attempts to predict the residual for voiced frames of speech, since the residual in unvoiced frames contains

very little information about the nature of the speaker, as there is no vocal fold activity. A GMM with Q_{rp} components was fitted to the CC's for the voiced frames of data. For each component of the GMM a codeword was calculated. This codeword has a magnitude spectrum, which was computed by summing the magnitude spectra of all the residuals, weighted according to the probability of each datapoint (frame of Cepstral Coefficients) belonging to that component. If $h_{q,i}$ is the posterior probability of C_{train} (the training data) for a class q and frame i, then the magnitude for codebook entry q is:

$$m_q = \sum_{i=1}^{N} M_i . \frac{h_{q,i}}{\sum_{j=1}^{N} h_{q,j}}$$

The codeword also contains a table of all the phases of the frames which have a 90% or greater probability of belonging to that component. The value of 90% was chosen in order to ensure there was a large enough number of entries in the table to provide reasonable spread of lengths of the associated phases, which will be important for reasons explained in 5.2.

5.2. Residual Prediction

Given the set of cepstral coefficients associated with a voiced frame of speech, the residual may be predicted in the following way. The magnitude spectrum of the residual was computed by summing all the codeword magnitudes, weighted according to the probability of the datapoint belonging to the component that this codeword is associated with. This is $\widehat{M_i} = \sum_{q=1}^{Q} m_q . h_{q,i}$.

This method for predicting magnitudes has the desirable property of the resulting magnitude spectrum changing smoothly provided the input parameters change smoothly. This avoids many of the artifacts associated with vector quatization methods [1].

Unfortunately, the same approach may not be taken with the phase, since phase may not be interpolated using a weighted sum analogous to the magnitude summation due to the way in which phase may 'wrap around' (i.e. a phase of 2π is equivalent 0). Resampling a phase to be a different length also requires interpolation. Therefore, the phase was computed by finding the most likely component of the GMM and choosing the phase from the associated table that was closest in length to the desired frame length.

After a phase and magnitude vector had been obtained, the magnitude vector was resampled to be of the same length as the phase vector. The inverse Fourier transform was then used to convert the magnitude and phase back into a time-domain signal.

5.3. Transformation

In order to perform the transformation, we require a set of Cepstral Coefficients for each frame of speech. These may either be predicted using the method of section 4, or obtained directly from the target speech if the system is being used purely to do residual prediction. For each frame of speech, if the frame is voiced, then a residual is predicted on the basis of the Cepstral Coefficients of that frame. If it is unvoiced, then the source residual is used, though it is resampled to be of the correct length. It is acceptable to resample in this case, since the resampling process is being performed in the time domain rather than the complex frequency domain as was discussed earlier. Each frame of speech is resynthesised by filtering the residual using the appropriate LPC coefficients. Finally the speech is formed using the overlap and add method described in section 4.3.

6. Evaluation

6.1. Performance indices

6.1.1. Spectral envelope

The error between two aligned sets A and B of LSF vectors may be computed as the Euclidean distance between the two vectors. This is however not a useful way to evaluate the performance of a transformation system since it doesn't take into account the 'difficulty' of the mapping, i.e. the difference between the source and target vectors. The difference between two speakers is called the inter-speaker error $E_{LSF}(t(n), s(n))$, where $t(n)$ are the LSFs of the target speech. The LSFs of the predicted target speech are represented as $\hat{t}(n)$. The transformation error is the difference between the predicted and actual LSFs $(E_{LSF}(t(n), \hat{t}(n)))$. Kain suggested an LSF performance index P_{LSF} for assessing the quality of transformation in a voice transformation system, as follows:

$$P_{LSF} = 1 - \frac{E_{LSF}(t(n), \hat{t}(n))}{E_{LSF}(t(n), s(n))}$$

A value of $P_{LSF} = 0$ indicates that the output of the system is no more similar to the target than the source is, whereas a value of $P_{LSF} = 1$ indicates that the output of the system is identical to the target. In general, a higher value for P_{LSF} suggests a better system.

6.1.2. Spectral detail

In order to ascertain the relative effectiveness of the spectral detail transformation component depending on the parameter values used, it is necessary to have a method for measuring performance. The performance index P_{LSF} is not appropriate, since it only measures errors in LSFs. The most common measure used in speech coding tasks is the signal-to-noise ratio (SNR). We measure the SNR as:

$$SNR = 10.log_{10} \frac{\sum |FFT(s(n))|^2}{\sum (|FFT(s_c(n))| - |FFT(s(n))|)^2}$$

where $s(n)$ is the original speech, and $s_c(n)$ its coded form. The SNR of a whole utterance is computed by dividing the speech into a number of fixed length (20ms) frames, and then finding the average SNR of these frames, rather than simply finding the SNR of the whole utterance. A frame based approach better reflects the perceptual quality as errors in quiet and loud segments of the speech are computed separately. The error is computed on the magnitude spectrum, since this better reflects perceptual quality, as the human auditory system is not very sensitive to changes in phase. Higher SNR values indicate a better system.

6.2. Results

6.2.1. Spectral envelope

As previously discussed, the experiments were carried out on two pairs of speakers, with each speaker used once as source and once as target. Example output of our system may be found online [8]. Further detail can be found in [5].

A large number of different experiments were carried out in order to discover the effects of varying various parameters. The

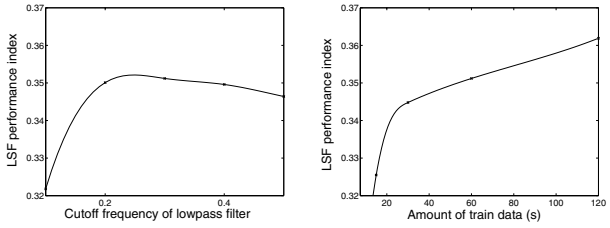

Figure 1: Left: Graph showing the relationship between the cutoff (as a fraction of the Nyquist frequency) of a lowpass filter applied to the LSFs (F_{LP}) and the performance of the resulting system (P_{LSF}). Right: Graph showing the relationship between the amount of training data (T_{train}) and performance (P_{LSF}).

variables in the following experiments are as follows: order of LPC analysis O_{LPC}, the number of components in the GMM Q_{LSF}, the cutoff frequency of a low pass filter applied to the transformed LSFs F_{LP} and finally the amount of training data T_{train}. The results were averaged for all four combinations of source and target speakers.

When we investigated the effect of changing the number of components in the GMM, it was found that a value of $Q_{LSF} = 12$ provided the best performance. The performance does not improve when there are more than 12 components of the GMM, and this is the case regardless of the amount of data trained on. We found that a value of $O_{LPC} = 20$ provides the best performance. This is consistent with the order of LPC analysis used in similar tasks [7].

The relationship between the cutoff of the low-pass smoothing filter and the performance of the resulting system is shown in figure 1 (left). The optimum cutoff is a value of $F_{LP} = 0.3$. The performance when doing smoothing with an appropriate cutoff value is substantially higher than the performance with no smoothing. This indicates that the smoothing plays a key role in obtaining good performance from the system. The movement of LSFs in natural speech is quite smooth. However, the transformation system works on a frame by frame basis resulting in noisy tranformed LSFs. Therefore, if there is too little filtering the transformed LSFs are still too noisy, and if there is too much filtering then information is lost. Figure 1 (right) shows that as the amount of training data is increased, the performance of the system improves. The largest amount of data used for training was 120 seconds which provided a value of $P_{LSF} = 0.36$.

6.2.2. Spectral detail

The SNR of the system varies with the number of components (Q_{rp}) in the residual prediction GMM. The highest SNR values (3.09 dB) are obtained when $Q_{rp} = 64$. Our results show that the SNR is lower when both the LSFs and residual are predicted (2.14dB rather than 3.09dB). This is of course what one would expect. The results show that as the amount of training data increases, the SNR also increases.

7. Conclusions

The results show that in order to gain the best performance, the following parameters should be used: $Q_{LSF} = 12, O_{LPC} = 20, F_{LP} = 0.3, T_{train} = 120s$. This leads to a performance of $P_{LSF} = 0.36$. In existing research, the voice transformation system which has the highest performance is a system by Kain [7], which has a performance of $P_{LSF} = 0.31$. Therefore, our system outperforms this system. Unfortunately we do not have access to the same test and training data as Kain used. Kain does not give the duration in seconds of the training data used, however he does state that 40 TIMIT sentences were used (i.e. 160 seconds). The training data used in our experiment is significantly different since it is prosodically varied. Kain asked the speakers to speak in a monotone, and to mimic the F0 contour, segment and word durations of a particular speaker to minimize intra-speaker error. It is easier to make the transformation if the timing and F0 are similar, since it is easier to find a good alignment, and also because the F0 of the speech does not need to be altered so much. Our system improves over Kain's system since it is able to deal with a more difficult problem: natural, prosodically varied speech. The improved performance index of our system over Kains could be due to the fact that our system rejects poorly aligned data from the training set, and also be due to the smoothing applied to the mapped LSFs.

It was found that when residual prediction alone is performed, the quality of the speech is extremely high, and it is quite hard to tell from the original speech. Example files may be found online [8]. When LSF mapping and residual prediction are performed, the quality is also good and may easily be recognised as the target speaker. However, there are artifacts typical of RELP manipulation.

8. References

[1] Levent M. Arslan and David Talkin, "Voice conversion by codebook mapping of line spectral frequencies and excitation spectrum," in *Proc. Eurospeech '97*, 1997, pp. 1347–1350.

[2] Y. Stylianou, O. Cappe, and E. Moulines, "Statistical methods for voice quality transformation," in *Proc. EUROSPEECH*, 1995.

[3] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University radio news corpus," Tech. Rep., Boston University, SRI International, MIT, 1995.

[4] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Signal Processing Series. Prentice-Hall, 1978.

[5] Ben Gillett, "Transforming voice quality and intonation," M.S. thesis, Centre for Speech Technology Research, University of Edinburgh, 2003.

[6] G. Baudoin and Y. Stylianou, "On the transformation of the speech spectrum for voice conversion," in *Proceedings of the International Conference on Spoken Language Processing*, 1996, pp. 1405–1408.

[7] Alexander Kain, *High Resolution Voice Transformation*, Ph.D. thesis, OGI School of Science and Engineering, October 2001.

[8] Ben Gillett, "Audio examples to accompany 'transforming voice quality and intonation'," http://www.cstr.ed.ac.uk/projects/voicetransformation, 2003.