TOWARDS A VOICE CONVERSION SYSTEM BASED ON FRAME SELECTION

T. Dutoit[1], A. Holzapfel[2], M. Jottrand[1], A. Moinet[1], J. Pérez[3] and Y. Stylianou[2]

[1]Faculté Polytechnique de Mons - BELGIUM [2]University of Crete - GREECE
[3]Universitat Politècnica de Catalunya - SPAIN

ABSTRACT

The subject of this paper is the conversion of a given speaker's voice (the source speaker) into another identified voice (the target one). We assume we have at our disposal a large amount of speech samples from source and target voice with at least a part of them being parallel. The proposed system is built on a mapping function between source and target spectral envelopes followed by a frame selection algorithm to produce final spectral envelopes. Converted speech is produced by a basic LP analysis of the source and LP synthesis using the converted spectral envelopes. We compared three types of conversion: without mapping, with mapping and using the excitation of the source speaker and finally with mapping using the excitation of the target. Results show that the combination of mapping and frame selection provide the best results, and underline the interest to work on methods to convert the LP excitation.

Index Terms— Voice conversion, frame selection, voice mapping

1. INTRODUCTION

Voice conversion has many applications in concatenative speech synthesis. This is especially true for systems using unit selection. In these systems, large databases should be developed for increasing the probability to obtain (on average) good quality of synthesis. Creation of large databases, however, is a time consuming task and very expensive (i.e., use of talent voices). Therefore, in this context, voice conversion is an attractive solution. Assuming that a database with a reference voice already exists, new voices (target voices) may be generated by applying voice conversion algorithms on the reference voice. One of the first voice conversion system proposed by Abe et al. [1] was based on Vector Quantization (VQ). The basic idea of this technique is to make mapping codebooks which represent the correspondence between two speakers. To avoid the limitations of the discrete space represented by VQ, a Fuzzy Vector Quantization (FVQ) has been proposed by Kuwabara et al. [2]. A method for *mapping* one class from the VQ space of the source speaker to the *corresponding* class in the VQ space of the target speaker has been proposed by Valbret et al. [3] based on Linear Multivariate Regression (LMR). In the same communication, Valbret et al. proposed a spectral transformation approach based on Dynamic Frequency Warping (DFW). The LMR approach proposed a simple linear transformation function for each class, while the DFW approach used a third order polynomial. All the above methods have been developed in the context of VQ. Mapping functions have also been proposed using more robust, modeling of the acoustic space of a speaker using a Gaussian Mixture Model (GMM). Assuming that source and target vectors obtained from the speakers acoustic space are jointly Gaussian, Stylianou et al. [4] have proposed a continuous probabilistic mapping function based on GMM. A similar mapping function has been proposed by Kain et al. [5] modeling jointly the source and target vectors with GMM. A different approach has been proposed by Iwahashi et al. [6] using speaker interpolation. All the above techniques are based on parallel training data, where both the source and target speaker utter the same sentence. In this case, Dynamic Time Warping (DTW) is used to time align the two signals, in order to extract matching source and target training vectors (i.e., Line Spectrum Frequencies, Mel Frequency Cepstrum Coefficients, etc.). Approaches without the requirement of parallel data have also been proposed in the literature [7] [8]. However, using the same mapping functions for parallel and non-parallel data, it has been shown that training with parallel data provides better conversion results [7].

In this paper, we try to design a voice conversion system from *x* (the source speaker) to *y* (the target speaker). It is assumed that a large amount of studio-quality speech data is available from the source and from the target. Our system is based on two independent blocks:

- mapping from *x* to *y'* (a first approximate of *y* using the parallel corpus, and
- speech-to-speech synthesis from *y'* to *y"* (a second and more accurate approximate of *y*).

The first block involves aligning the data on a frame by frame basis, and building a GMM-based mapping function. In the second block, for each frame in *y'*, we select a new frame from the target database. Our system presents some similarities with the segmental speech coder presented by Lee and Cox in [9], but we use, as basic units, frames instead of variable length segments.

This paper is organized as follows: Section 2 presents how we aligned the data for the training and presents the GMM-based mapping function. Section 3 presents the frame selection algorithm. The next section is devoted to the details of the three conversion systems we have tested and compared. Finally, sections 5 and 6 presents the results we obtained, conclusions and possible future work.

2. VOICE MAPPING

2.1. Data alignment

Although the corpus used for the voice mapping part of this project consists of parallel utterances, some timing differences are unavoidable due to different speaker characteristics. Since the training of the mapping function requires parallel data vectors, an alignment turns out to be necessary. This alignment has been produced by a dynamic time warping procedure. An iterative procedure has been applied to improve the alignment, by reapplying the DTW method between

This work is supported by the NoE SIMILAR within the 6th FP

converted and target envelopes [4]. After each iteration, a new mapping function can be estimated between the newly aligned original source and target data.

2.2. Voice mapping

When converting the voice of the source to the voice of a the target speaker we assume that these two voices are defined by their spectral spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively. Our problem in voice conversion is two-fold: at first we have to find a way to model these spaces and then we have to find a way to map a previously unknown example from the source space to the target space. In order to be able to find such a mapping we assume that there is aligned training data available. This means that we have two sets of spectral vectors $\mathbf{x}_t$ and $\mathbf{y}_t$ that describe spectral envelopes from source and target speakers respectively. The two sets of vectors $\{\mathbf{x}_t,\ t = 1, \ldots, N\}$ and $\{\mathbf{y}_t,\ t = 1, \ldots, N\}$ have the same length N and are supposed to describe sentences uttered in parallel by source and target. What is desired is a function $\mathcal{F}()$ such that the transformed envelope $\mathcal{F}(\mathbf{x}_t)$ best matches the target envelope $\mathbf{y}_t$, for all envelopes in the learning set ($t = 1, \ldots, N$).

To achieve the goal of voice conversion, we use a gaussian mixture models approach described in [4], which models the probability of a given vector X in the input space as:

$$p(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i) \tag{1}$$

where M is the number of Gaussians, $\mu_i, \mathbf{\Sigma}_i$ are the mean vector and the covariance matrix of the i-th Gaussian component, and α_i are the weights used to combine the different components. These M Gaussian components can be regarded as classes within the spectral space of the source and a vector $\mathbf{x}_t$ can be classified to one of the classes using maximum likelihood. The mapping to the target space is done by using these parameters in the conversion function

$$\mathcal{F}(\mathbf{x}_t) = \sum_{i=1}^{M} P(\mathcal{C}_i|\mathbf{x}_t) \left[\nu_i + \mathbf{\Gamma}_i \mathbf{\Sigma}_i^{-1}(\mathbf{x}_t - \mu_i)\right] \tag{2}$$

where ν and $\mathbf{\Gamma}$ are related to the mean target and the cross-covariance matrix of the source and target vectors. The conversion function parameters are determined by minimization of the total quadratic spectral distortion between the converted envelopes and the target envelopes:

$$\epsilon = \sum_{t=1}^{N} ||\mathbf{y}_t - \mathcal{F}(\mathbf{x}_t)||^2. \tag{3}$$

For details on the minimization see [4].

3. FRAME SELECTION

Once the features of the frames of the original speaker have been converted using the GMM mapping conversion, the new vectors of features $Y' = [y'^{(1)} \ldots y'^{(t)} \ldots y'^{(T)}]$ are used as inputs of the frame-selection algorithm.

The principle of this algorithm is similar to the unit-selection algorithm used in concatenative speech synthesis, using dynamic programming (Viterbi) to select units in a database. However, TTS systems usually deal with units such as diphones, phones or parts of phones while our algorithm uses smaller units : 32 ms frames (with a constant shift of 8 ms between each frame).

3.1. Clustering

First we use the LBG method [10] to divide the target database into 256 clusters. The centroid of each cluster is the mean value vector of the features of the frames inside that cluster.

Then for each set of new features $y'^{(t)}$ we select the cluster with the closest centroid. The closeness of a frame to a centroid is measured with a weighted euclidean distance

$$closest\ centroid = \underset{c=1,\ldots,256}{\arg\min} \sum_{i=1}^{L} w_i \cdot \left(y_i'^{(t)} - \bar{y}_i^{(c)}\right)^2, \tag{4}$$

where L is the dimension of the feature vectors, $y_i'^{(t)}$ is the i^{th} component of the t^{th} feature vector produced by the mapping function, $\bar{y}_i^{(c)}$ is the i^{th} component of the c^{th} centroid of the database and w_i is the weighting factor associated with these i^{th} components.

As a result, we obtain a sequence of clusters $[c(1) \ldots c(T)]$ to be used in the next step of the algorithm, the frame selection.

This pre-selection of groups of frames (clusters) reduces the number of candidates for each step of the Viterbi algorithm (see 3.2), thus reducing efficiently the computational time while not affecting the quality of the final output.

3.2. Frame selection

We use the Viterbi algorithm to select a sequence of frames $Y = [y^{(1)} \ldots y^{(t)} \ldots y^{(T)}]$ from the target database, minimizing a distance (the *overall distance*) between those frames and the output sequence Y' of the mapping function. The frames Y will be chosen within some target speech database large enough to see all sorts of acoustic events (see section 5 for detailed information),

The overall distance is a combination of target and concatenation costs. At frame t, the target cost (t_{cost}) is the weighted euclidean distance between the vector of converted features $y'^{(t)}$ and the features of one frame among the $M_{c(t)}$ frames belonging to the cluster $c(t)$

$$t_{cost}(t, m_{c(t)}) = \sum_{i=1}^{L} w_i \cdot \left(y_i'^{(t)} - y_i^{m_{c(t)}}\right)^2 \tag{5}$$

Likewise, the cost of concatenation (c_{cost}) is the weighted euclidean distance between a frame of cluster $c(t)$ and a frame of cluster $c(t+1)$

$$c_{cost}(m_{c(t)}, m_{c(t+1)}) = \sum_{i=1}^{L} w_i \cdot \left(y_i^{m_{c(t+1)}} - y_i^{m_{c(t)}}\right)^2 \tag{6}$$

Moreover, the selection process is biased towards favoring, for consecutive frames, the selection of consecutive frames from the speech database by setting the concatenation distance to zero in this case. This will advantage the selection of neighbour frames in order to reduce discontinuities during the synthesis of speech.

The frame selection process also prevents the same frame from being selected two times in a row.

This approach is very similar to that developed in Suenderman *et al.* [8], with the difference that in our approach the target sequence for the frame selection algorithm is the mapped sequence Y' while Suenderman *et al.* use the input sequence X as target.

In order to increase the speed of the algorithm, we also selected for each frame $y'^{(t)}$ the N closest frames (in the sense of target cost) inside cluster $c(t)$. Only those frames were used in Viterbi as candidates.

4. SPEECH SYNTHESIS

We used the previously described algorithms in three different settings for voice conversion : the first setting does not use mapping, and produces speech by LP synthesis using the source LP excitation and the LP filter obtained by frame selection ; the second setting add a mapping step; the third setting uses the LP excitation of the target.

4.1. Method a: no mapping, source LP excitation

The first case is illustrated in figure 1. For each frame of the source, MFCC vectors are extracted. Without any mapping, these MFCC vectors are directly used to select the best matching frames (with the Viterbi algorithm described in section 3). The Viterbi algorithm gives as output an MFCC vector containing the MFCC of the frame selected but also information on how to locate each of the frame selected. With this information, one retrieves the selected frames and an LPC analysis is achieved to produce the auto-regressive (AR) coefficients. In parallel, one uses an LPC inverse filter to get the excitation of the source. The converted speech is then synthesized using this excitation and the AR coefficients.

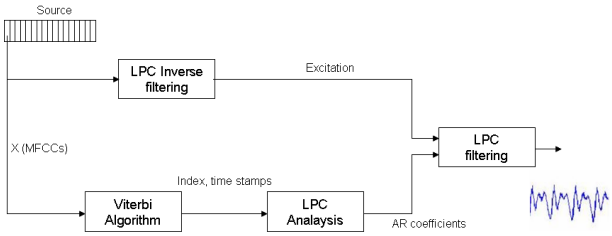

Fig. 1. Source voice resynthesis by frame selection

4.2. Method b: mapping, source LP excitation

Figure 2 shows a more realistic voice conversion system. Here the MFCC vectors extracted from the source are mapped to get corresponding Y' MFCC vectors. The new features Y' are then used as input of the frame selection system. The converted speech is produce by LP analysis - synthesis using the excitation of the source exactly in the same way as in the first case.

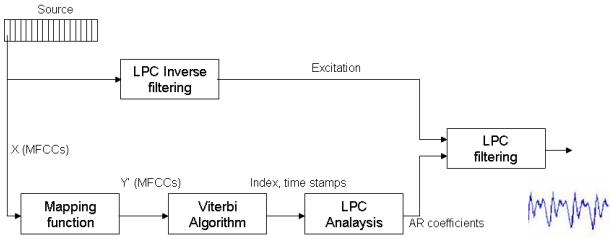

Fig. 2. Voice conversion using the source excitation

4.3. Method c: mapping, target LP excitation

The third setting, illustrated in figure 3, is similar to the previous one, but, instead of using the source LP excitation, we use the target one. This requires, before the frame selection, an alignment from the source to the target, in order to get the right number of source MFCC vectors, since the number of frames of the source utterance and target utterance are different. The alignment is achieved by a DTW algorithm. The aligned MFCC vectors are then used as input of the frame selection block, while in parlallel, a LP inverse filtering extracts the target excitation. The converted speech is synthesize as previously.

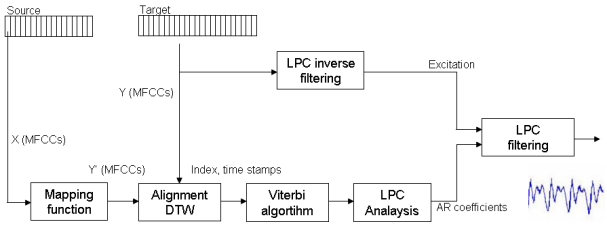

Fig. 3. Voice conversion using the target excitation

5. EVALUATION RESULTS

The target database in which the selected frames are extracted contains 93212 frames from a few hundred target speaker utterances. We used 20 MFCC and the LPC analysis and synthesis order is also 20.

We have performed a formal evaluation of the three proposed methods. The test is based on subjective rating by human judges (MOS test). 19 judges were recruited, non-English native speakers with no known hearing problems. All the judges were familiar with speech processing techniques, although only a few were experts in speech synthesis. The test was carried out using a web-based setup. Following the common approach when evaluating Voice Conversion techniques, two different metrics were used in the evaluation: one for rating the success of the transformation in achieving the desired speaker identification, and one for rating the quality of the transformed speech. This is required since the changes required to obtain a high degree of similarity usually introduce large distortion, thus degrading the quality of the output signal. For the similarity test we chose the well-known AB test: the listeners were presented with two utterances, one resulting from the transformation and another (different) from the target database. Then, they were asked to decide if the voices came from different speakers (1) or from the same speaker (5), using a 5-points scale. Some source-target examples were introduced for reference purposes. In the quality test, the human judges were presented with examples of the different transformation techniques, and were asked to rate the quality of each example using a 5-points scale, from bad (1) to excellent (5). Again, some natural examples were introduced as a reference both for the listeners and for subsequent interpretation of the results.

Table 1 presents the results of the two MOS tests. Method ***c*** shows the highest similarity score, followed by methods ***b*** and ***a***, coinciding with the ranking obtained computing the distance from the converted frames to the target frames as presented in table 2 (these distances correspond to the averaged euclidean distances between MFCC vectors; if the files to be compared have not the same length, the MFCC vectors are aligned by a DTW algorithm). This is to be expected, since the later two methods (***a*** and ***b***) work with target LPC filters, maintaining the source excitation. On the other hand, method ***c*** selects the excitation from the target database, achieving a higher identification score. However, none of the three methods is judged to produce *similar* voices to the target examples. From the quality

Method	Similarity	Quality
a	1.82	3.71
b	2.29	3.52
c	2.77	2.56
Source	1.71	4.95
Target	4.53	4.82

Table 1. Results of the evaluation of the three proposed methods: MOS test evaluating the similarity between two utterances (*converted* and *target*), MOS test evaluating the quality of the *converted* utterance. For reference purposes, the rows labelled *Source* and *Target* present the same measures when provided with natural source and target sequences

Source-Target	33.08
Method a - Target	30.39
Method b - Target	26.27
Method c - Target	23.61

Table 2. Average objective distances

test, the most successful methods are ***a*** and ***b***, both rated to be Fair-Good. Not surprisingly, the larger modifications come at the penalty of decreasing the quality of the resulting waveform: method ***c*** is not rated to be of acceptable quality (Bad-Fair).

As it can be seen, more work is required in order to achieve a higher identification score, since none of the three methods has proved successful in that test. However, some important conlusions can be drawn :

1. Adding mapping before frame selection (setting b vs setting a) provide a shift in similarity. As a matter of fact, using the *source* MFCCs as input to the frame selection algorithm (i.e., not using any mapping, as in [8]) simply provides the closest MFCCs from the target database. This selection has little chances of having an important voice conversion contribution. In practice, the output of test a even often sounded close to the source itself!
2. Using target excitation as input to the LP synthesis filter results in a shift in voice similarity (seting c vs. setting b). This is of course not astonishing, since the target excitation contains a lot of the voice quality of the target voice, which is know to be an important cue for speaker identification. This indicates that some efforts should now be allocated to the conversion of source to target excitation, which we consider as the real challenge.

This will also require some research in order to avoid penalising the quality of the resulting voice.

6. CONCLUSIONS AND PERSPECTIVES

In this paper, we present a frame selection method for voice conversion. We use a mapping function that maps MFCC vectors from a source speaker into those of a target speaker. These mapped MFCC vectors are then used to select, with a Viterbi algorithm, real frames in a database containing frames from the target speaker. We synthesize speech by LPC analysis and synthesis. Three different cases were studied: one without any mapping, and using the source LPC residual, one with mapping using also the source residual and finally one with mapping and using the target residual. Our results show that it is possible to obtain a good degree of similarity by using the target LP excitation as input to the LP filter, whose coefficients are obtained from frames selected from the target database, after mapping from frames in the input utterance. Further work should be devoted to converting source excitation into target excitation. This can somehow be done independently of the spectral envelope selection proposed in this paper.

7. ACKNOWLEDGMENTS

This work is a follow-up from the Multimodal Speaker Conversion project from eNTERFACE06, the Summer Workshop organized by the SIMILAR network (www.enterface.net/enterface06). We want to thank Ferda Ofli, Ferran Marques and Antonio Bonafonte for their contribution in the project.

8. REFERENCES

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. ICASSP88*, 1988, pp. 655–658.

[2] H. Kuwabara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: Control and conversion," *Speech Communication*, vol. 16, no. 2, pp. 165–173, 1995.

[3] H. Valbret, E. Mulines, and J.P. Tubach, "Voice transformation using PSOLA techinques," *Speech Communication*, vol. 11, no. 2.

[4] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, vol. 6, no. 2, pp. 131–142, 1998.

[5] A. Kain and M. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. ICASSP98*, 1998, pp. 285–288.

[6] N. Iwahashi and Y. Sagisaka, "Speech spectrum transformation based on speaker interpolation," in *Proc. ICASSP94*, 1994.

[7] A. Mouchtaris, J. Van derSpiegel, and P.Mueller, "Non parallel training for voice conversion based on a parameter adaptation," *IEEE TRANSACTIONS ON SPEECH AUDIO and LANGUAGE PROCESSING*, vol. 14, no. 3, pp. 952–963, 2006.

[8] D. Suendermann, H. Hoege, A. Bonafonte, H. Ney, A. Black, and S. Narayanan, "Text-independent voice conversion based on unit selection," in *Proc. ICASSP06*, Toulouse, 2006, pp. 81–84.

[9] R.V. Cox K. Lee, "A segmental speech coder based on a concatenative tts," *Speech Communication*, vol. 38, no. 1, pp. 89–100, 2002.

[10] A. Gersho and R. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, Norwell, Massachusetts, 1992.