VOICE CONVERSION BASED ON TOPOLOGICAL FEATURE MAPS AND TIME-VARIANT FILTERING

Ansgar Rinscheid

Lehrstuhl für allgemeine Elektrotechnik und Akustik, Ruhr-Universität Bochum
D-44780 Bochum, Germany, e-mail: rinsch@aea.ruhr-uni-bochum.de

ABSTRACT

This paper presents a new voice conversion algorithm. This algorithm allows voices to be adapted using a small amount of adaptation data. Only a few short adaptation units (phonemes or short words) are needed. The voice conversion is performed using a time-variant digital filter, topological feature maps and a map of filter coefficients. The filter coefficients of the time-variant filter are selected by the feature map dependent on the short-time spectrum. The spectral envelope of the input signal is modified by a time-variant filter using the selected coefficients.

1. INTRODUCTION

Various techniques of speaker adaptations have been developed to reduce the difference in performance between speaker-dependent and speaker-independent recognition systems. Most of the adaptation algorithms transform the speech recognition system itself or the input data (feature vectors), but they do not transform the wave form of the speech signal. Therefore it is very difficult to make the adaptation results audible.

An adaptation algorithm which uses speech as an input and generates a new speech signal with different properties (speech-in-speech-out) has to preserve the speech quality of the input signal. Noise or other distortions introduced by the adaptation procedure are not acceptable, if the adaptation is to be used for speech synthesizers in dialog systems. A speech-in-speech-out speaker adaptation system can be used for speech recognition and speech synthesis applications, because the wave form of the speech signal is transformed.

Many new time domain speech synthesizers generate high quality synthetic speech. The speech is generated by concatenating speech units (diphones, demisyllables, ...), which are recorded from a human speaker. Consequently, the resulting sound of the synthetic speech only depends on the speaker. If a different synthetic voice is needed, all the speech units must be recorded once more with another speaker. In the case of a voice conversion algorithm the sound of the synthetic voice can be modified and it is not necessary to record all the speech units again.

Voice conversion is especially interesting in translation systems in multi-party scenarios. In this case the aim is that the voice of the translated speech should sound like the input speech. A female voice does not have to be translated with a male voice. If the translation system is used from three or more speakers at the same time, it must be possible to identify each speaker by means of the corresponding synthetic voice. A translation system requires an on-line-adaptation algorithm, which works in real time. This requirement makes the adaptation task especially difficult.

Some existing voice conversion algorithms use neural networks or vectorquantisation in order to select linear transformation rules. The transformations can then be performed by shifting formant frequencies or other spectral characteristics [5] [6].

This paper presents an algorithm used in order to convert the spectral envelope of speech signals. This algorithm allows speech sound to be modified using a small amount of adaptation data. Only some short adaptation units (phonemes or short words) are needed. The spectral envelope is only a small part of the speech information. Other parts are prosodical informations such as fundamental frequency contour, intensity and rhythm. Therefore the aim of the presented method cannot be to imitate a speaker's voice, but to modify of the speech's sound.

2. THE VOICE CONVERSION ALGORITHM

The voice conversion algorithm is based on a set of linear transformation rules, which are selected according to the spectral features of a short-time signal (st-signal) in an operation phase.

The selection is done by choosing the winner of the feature map. The feature map performs a vector quantization which subdivides the feature space into a fixed number of subspaces. Each neuron of the feature map represents one subspace. The feature map is self-organizing in a training phase, using the feature vectors of one speaker [4]. Thus, the spectral features, and the location and shape of the subspaces are usually speaker dependent. The self-organizing feature map is used as the reference map.

To create a system which is almost speaker independent, one map for every different speaker has to be determined in an adaptation phase. These maps, called test maps, are trained by a modified 'Forced Competitive Learning' procedure to ensure the topological identity of the maps and thereby they implicitly establish a 1:1 correspondence to the code books [1][2][3]. This allows a 1:1 mapping of the feature vectors represented by the reference map and the test maps. The presented algorithm uses the reference map and the test maps to determine the adaptation rules for the mapped neuron. The

adaptation rules describe how st-signal features in a subspace of a certain speaker have to be converted in order to move into the corresponding subspace of a different speaker (Fig. 1).

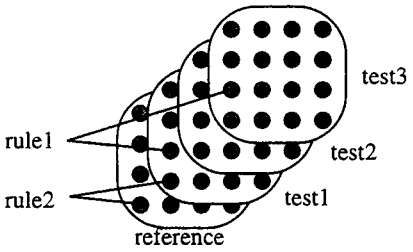

Figure 1: The reference and three test feature maps are shown. The grey points represent the neurons. The rules (rule1, rule2) describe the adaptation rules between two corresponding neurons from different maps. The overall voice conversion procedure is divided into six steps:

2.1. Extraction of features from the speech signal

In order to extract the features the speech signals are windowed pitch synchronously with the 'hanning-window'. The length of the window is 32 ms. The st-signals are transformed in the frequency domain using the FFT-Algorithm (Fast-Fourier-Transformation). The feature vector has 30 dimensions. The elements V_i of the feature vector are calculated according to the following equation:

$$V_i = \left(\frac{1}{N-M} \cdot \sum_{n=M}^{N-1} \left| c(n) \cdot w_i(n) \right|^2 \right)^{0.15}$$

Where M denotes the index of the first and N-1 the index of the last value in the window w_i, c(n) are the complex values of the FFT-spectrum. The weighting functions w_i are illustrated in Fig. 2

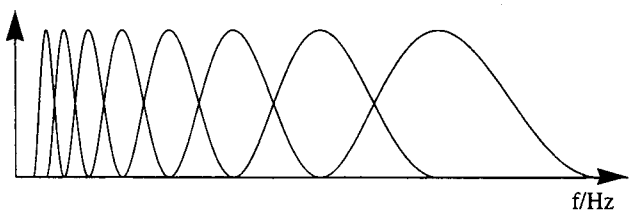

Figure 2: Spectral weighting windows (frequency bands).

The length of each weighting window is 60 mel (Bark-scale).

2.2. Training of the reference feature map

The basis of the proposed voice conversion method is a self-organizing feature map trained on the feature space of the reference speaker's adaptation data (reference map). The map size is quite small and dependent on the adaptation data. Approximately two or three neurons per phoneme represented by the adaptation data are used. The training is performed by applying the well-known competitive learning rules of T. Kohonen [4]. Training the feature map is the most time consuming step in the conversion procedure. Note that the training is carried out off-line and only once per adaptation data. Thus, the computational time necessary can be neglected.

2.3. Determination of a test map

At this stage a map representing the test speaker's feature space with the same topology as the reference map is created. A feature map obtained by implementing a self-organizing training algorithm probably has a different topology. Therefore the test map is not trained, the neurons of the map are set directly in an adaptation procedure. First, the adaptation data (same utterances from the reference and test speakers) are aligned using dynamic time-warping or phoneme segmentation. In the latter case a linear time-warping for each phoneme is performed. Then the following steps are executed for all the feature vectors:

1. Take the feature vectors of the reference speaker and find the winners on the reference map.

2. Adapt the feature neuron at the winner's location (reference map) on the test map to the corresponding feature vector of the utterance made by the test speaker.

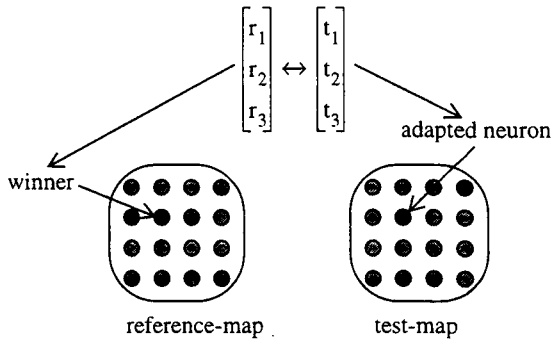

Figure 3: Training of the test map. The vector r and t denote the feature vectors of the reference and test speaker.

The training of the test map is in principle similar to the training of the reference map in that a winner is determined and adapted to a feature vector. But in the proposed training algorithm the winner of a reference vector is determined on the reference-map and the test-map is modified according to the corresponding feature vector of the test speaker. At the end of this procedure the neurons of the test map are set to the mean input vector associated to the adapted neurons.

2.4. The short-time signal map

After the reference map and one or more test maps have been created, one st-signal map is generated for each feature map. The st-signal map contains the most representative st-signals of one speaker. For every neuron the st-signal with the smallest quantization error when compared with feature neuron is taken from the adaptation data. The visualization of two st-signal maps is shown in

the figure below.

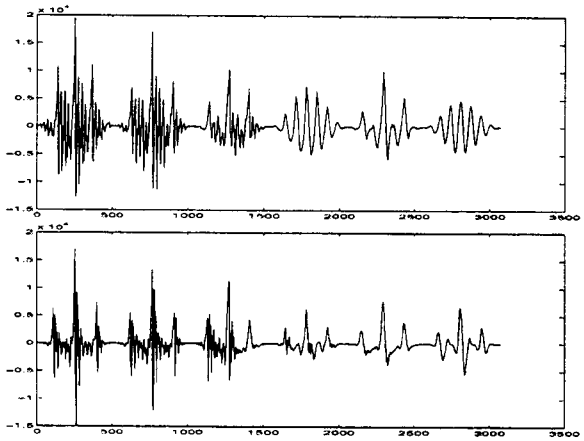

Figure 4: This figure shows the st-signals of two st-signal maps. The map size is 6x1 st-signals (6 neurons). The first plot represents the st-signals of the reference [A235S7.WAV] and the second of the test speaker [A235S8.WAV].

2.5. The transformation map

Based on two st-signal maps a transformation map is created using the 1:1 mapping feature of the corresponding feature maps. In the current approach a two folded LPC-analysis is performed for all the mapped st-signals in order to compute the transformation rules. The linear transformation is performed by a digital filter. The filter coefficients of the time-variant filter are determined in the following steps:

1. Calculate the PARCOR coefficients (reflection coefficients) using the st-signals of the reference map (for instance by using the covariance- or burg-algorithm).

2. Filter the st-signals of the test map with the calculated PARCOR coefficients. The spectrum of the resulting residual signal is an estimation of the difference spectrum of the st-signals. If the two used st-signals have the same spectral envelope, the spectrum of the residual signal appears to be white (whitening feature of the LPC-analysis).

3. Calculate the PARCOR coefficients of the residual signal. The PARCOR coefficients calculated in this way estimate the filter, which is used to convert the spectral envelope.

The filter can used as a FIR- (Finite Impulse Response) or IIR-filter (Infinite Impulse Response) depending on the direction of the conversion. If the FIR-filter converts the spectrum of the reference speaker to the test speaker, the corresponding IIR-filter can be used to convert the test speaker's voice to the reference speaker's voice.

2.6. Voice conversion with time-variant filtering

In the last step the voice is converted using the time-variant digital filter. The filter coefficients are selected according to the spectral features by using the speaker's feature map and the filter calculated coefficients. At first, the winner is determined on the feature map, and then the filter coefficients at the winner location are selected for the transformation.

The feature extraction and classification is very time consuming (not real time), but in the case of a speech synthesizer the speech signals (synthesis units) are known, and the classification of the st-signals can be done off-line. Only the time-variant filtering has to be performed on-line. To prevent clicks or other distortions in the converted signal, a lattice filter is used and the coefficients of the filter are interpolated sample by sample. Without an interpolation the filter becomes a parametric amplifier, if the coefficients change too much.

3. RESULTS

This chapter describes comparisons of converted speech signals in the time and frequency domain. The results are obtained by executing the described voice conversion algorithm. Only one word with two different phonemes is used as the adaptation data (adaptation data: /nananan/, reference voice [A235S1.WAV], test voice [A235S2.WAV], converted test voice [A235S3.WAV]). The map size is 6x1 neurons. The st-signal maps of the reference and test speaker (both male) can be seen in Fig. 4 (6x1 st-signals). The order of the digital filters (transformation map) is 50. The time-variant conversion of a test signal is performed with a time-variant fir-filter.

In the first experiment the adaptation data itself is converted. The results are shown in Fig. 5.

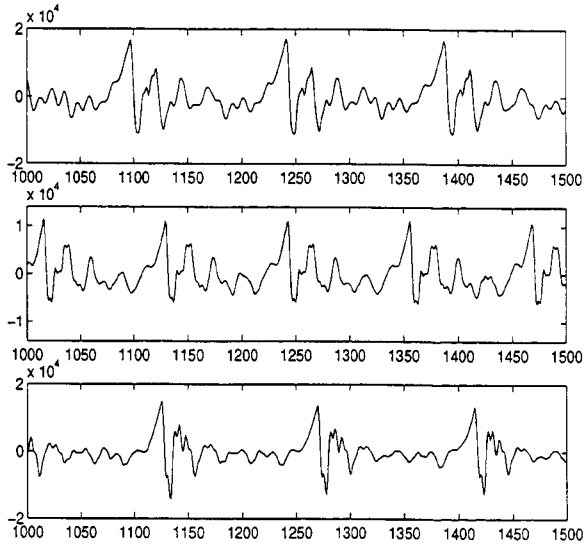

Figure 5: This figure represents adaptation results of an /a/ in the time domain. At the top the reference signal is presented [A235S4.WAV], and at the bottom the test signal [A235S5.WAV]. The converted test-signal is between these two signals [A235S6.WAV].

Evaluating the adaptation results in the time domain is difficult, but it can be seen that the wave form of the test signal has shifted towards the reference signal without visible distortions.

In the frequency domain the adaptation of the test speaker's signal towards the reference signals can definitely be shown (Fig. 6). The spectral envelopes of a converted /a/ from the test speaker become very similar to the phoneme from the reference speaker.

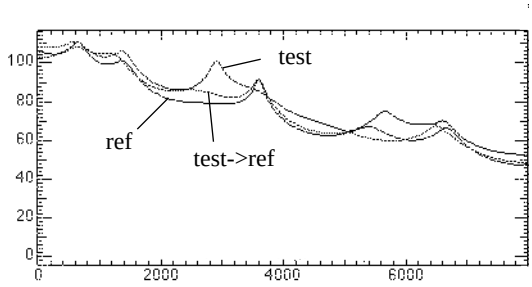

Figure 6: This figure represents the adaptation results in the frequency domain (log-power lpc-spectrum of the phoneme /a/). The results are obtained by using the burg method. (lpc-order = 20, window = rectangle).

Adapting the adaptation data is, of course, the easiest procedure, but the results show that adapting the speech signal /nananan/ based on only six adaptation rules is possible, and that the determined filter modifies the signal spectrum in the desired direction.

Any other signal can also be converted using the same procedure. Each st-signal must be labeled with the feature map of the speaker, and then the signal is transformed using the selected time-variant filter coefficients.

But what happens if an arbitrary sentence is converted using the adaptation rules based on only two phonemes (/nananan/)? The selection of a transformation rule is done with the feature map by determining the winner, therefore an st-signal is always converted with the rule of the most similar signal on the st-signal map.

Problems occur if the selected filter coefficients change rapidly. In order to reduce this effect, the filter coefficients are interpolated sample by sample nevertheless, in some cases distortions appear in the filtered speech signal.

In general, the energy of the output signal depends on the frequency response of the filter and the spectrum of the filtered signal. Thus, the intensity contour of the speech signal is not controlled by the algorithm. This causes, in some cases, rapid changes to the energy (Fig. 7).

4. CONCLUSION

The presented voice conversion algorithm converts the spectral envelope of speech signals. The algorithm is capable of working with only a few adaptation units. Investigations have to be carried out as to how the slight distortions caused by the transformation can be minimized and in what way the adaptation data influence the results. It is probable that an increasing number of adaptation units will improve the results.

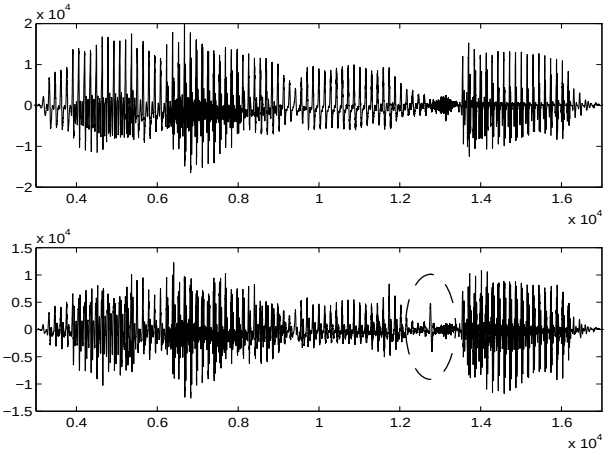

Figure 7: This plot displays the waveform of an original (top) [A235S10.WAV] and a converted (bottom) speech signal [A235S11.WAV]. A distortion introduced by the adaptation algorithm is marked with the ellipse (target signal [A235S9.WAV]).

ACKNOWLEDGMENT

This research was carried out as a part of the language&speech project VERBMOBIL supported by the German ministry of science and technology.

REFERENCES

1. Knohl, L., Rinscheid, A., "Speaker Normalization and Adaptation Based on Feature-map projection", Proc. EUROSPEECH-93, 3rd European Conf. on Speech, Communication and Technology: 367-370, 1993.

2. Knohl, L., Rinscheid, A., "Speaker Normalization with Self-Organizing Feature Maps", Proc. IJNN-93-Nagoya, int. Joint Conf. on Neural Networks: 243-246, 1993.

3. Knohl, L. & Rinscheid, A., "Verfahren zur gegenseitigen Abbildung von Merkmalssätzen", German Patent application P 43 00 159.9-53

4. Kohonen, T., "Self-Organization and Associative Memory", 3. Edition, Springer, Berlin, 1989.

5. Hideyuki Mizuno, Masanobu Abe, "Voice conversion algorithm based on piecewise linear conversion rules of formant frequency and spectrum tilt", Speech Communication 16: 153-164, 1995.

6. Narendranath, M., Murthy, H. M., Rajendran, S., Yegnanarayana, B., "Transformation of formants for voice conversion using artificial neural networks", Speech Communication 16: 207-216, 1995.