Real-time Voice Conversion Using Artificial Neural Networks with Rectified Linear Units

Elias Azarov, Maxim Vashkevich, Denis Likhachov, Alexander Petrovsky

Computer engineering department, Belarusian State University of Informatics and Radioelectronics,
6, P.Brovky str., 220013, Minsk, Belarus

{azarov, vashkevich, likhachov, palex}@bsuir.by

Abstract

This paper presents an approach to parametric voice conversion that can be used in real-time entertainment applications. The approach is based on spectral mapping using an artificial neural network (ANN) with rectified linear units (ReLU). To overcome the oversmoothing problem a special network configuration is proposed that utilizes temporal states of the speaker. The speech is represented using the harmonic plus noise model. The parameters of the model are estimated using instantaneous harmonic parameters. Using objective and subjective measures the proposed voice conversion technique is compared to the main alternative approaches.

Index Terms: voice conversion, artificial neural networks

1. Introduction

The aim of the paper is developing a voice conversion technique that can be used in voice over IP (VoIP) systems. The main requirements for voice conversion in this context are: real-time signal processing, high speech intelligibility, support of different sample rates of both source and target speech signals and scalable computational complexity.

Voice conversion implies mapping of speech parameters of the source speaker (spectral, excitation and prosodic features) to the acoustic space of the target speaker. The mapping rules are extracted during training which usually utilizes same parallel utterances of source and target speakers.

The statistical mapping approach is one of the most widely used in voice conversion. The approach is based on Gaussian mixture model (GMM) and has been extensively exploited in the last two decades [1-2]. The main disadvantage of the method is smoothing of spectral details because the transformed spectral envelopes are modeled as a mixture of their average representations. The problem has been partially solved by the recently proposed trajectory-based conversion method [2] that significantly improves conversion quality. The method has been implemented later as a postfiltering process specially designed for real-time conversion [3].

The other popular approach to spectral mapping, that can be used in real-time voice conversion, is frequency warping (FW) [4]. This technique is based on mapping the frequency axis of the source and target speakers' spectrum. Since the transformation of the frequency axis does not lead to any loss of spectral details the technique outperforms even the state-of-the-art GMM-based conversion in terms of perceptual quality of the output signal. Though the FW-based methods produce natural-sounding speech they are not as good in terms of similarity of the converted and target speakers.

Compared to GMM and FW-based methods ANNs are not very popular in voice conversion, though it has been shown that they can be as good as the state-of-the-art GMM-based techniques [5]. The following reasons can be given for such disregard: 1) high computational cost is required to train an ANN; 2) voice conversion quality is close to other approaches but not significantly better; 3) many of ANN structures are hardly interpretable in statistical terms; 4) the underlying signal model for GMM and ANN-based approaches possibly should be different.

The last and probably the most important reason for the modest success of ANNs in voice conversion is that the net configurations being used are essentially the same as for recognition and classification tasks (in [6] the radial basis and in [5,7] sigmoid units are used). However, unlike GMM in ANN-based conversion there is no appropriate substitution for global variance (GV). As the result the ANN-based conversion is prone to oversmoothing and sometimes produces unacceptable results.

The voice conversion technique described in this paper uses hybrid parametric periodic/aperiodic/mixed model based on instantaneous harmonic parameters. Unlike conventional approaches we use subband log energy values instead of mel-cepstral coefficients (MCEPs) for envelopes representation. This preserves natural correlation between source and target envelopes and simplifies the mapping task. The second advantage of such representation is independence of spectral lines that makes it easy to split the ANN into many independent nets of lesser capacity and simplify the training procedure. Besides that, it is easy to scale the mapping function to any combination of source and target sample rates.

The proposed neural network architecture uses temporal states of the speaker to overcome the oversmoothing problem. Considering that the mapping function is close to linear and that the output data are basically real-valued we use ReLU. It has been shown that this activation function has significant advantages over logistic both in supervised and unsupervised learning and can be efficiently applied for speech processing [8].

The proposed technique does not use any linguistic features (as well as GMM and FW-based methods). That makes it applicable for cross-language speaker conversion.

2. Basic algorithms

2.1. Feature extraction

In order to perform voice conversion the following features are extracted from the speech: spectral envelope, instantaneous pitch and excitation type that can be voiced, unvoiced or mixed.

The features extraction technique is based on harmonic plus noise model that provides a flexible and explicit control over speech parameters. The technique can be shortly summarized in the following way: 1) instantaneous pitch values are extracted using the instantaneous robust algorithm for pitch tracking (IRAPT) [9]; 2) the signal is transformed (warped) in time domain to get the signal with constant pitch; 3) instantaneous harmonic parameters are estimated using a DFT-modulated filter bank; 4) according to the estimated instantaneous frequencies the spectrum regions are classified as periodic or stochastic; 5) periodic components are synthesized and subtracted from the signal; 6) the residual is transformed into frequency domain using the short-time Fourier transform; 7) the estimated instantaneous harmonic parameters and the residual spectrum are combined into joint spectral envelope; 8) adjacent spectral envelopes are analyzed by excitation detector that makes the whole frame decision – voiced/unvoiced or mixed.

The joint spectral envelopes are represented as log energy values uniformly spaced in the mel scale. For the speech signal sampled at 44.1kHz we use 100 spectral components. This number is a tradeoff between reconstruction quality and computational complexity.

It is a common practice to decorrelate the envelope components using MCEPs that can significantly reduce vector's dimensionality and simplify GMM-based training. We do not use MCEPs because of a different learning approach. Unlike GMM-based learning, which is basically a soft classification task, we are trying to find a direct dependence of outputs on inputs which is a regression task. In this case conversion of decorrelated vector sequences would require much more powerful mapping models with more pronounced nonlinearity.

According to its purpose the voice conversion system is designed to handle wide range of input and output sample rates. Moreover the training and conversion can be done on different sample rates. As long as envelope components are independent energy values it is relatively easy adapt the solution to such conditions.

2.2. Alignment of parallel utterances

Data alignment is performed using iterative dynamic time warping (DTW) with linear regression (LR) [10]. At each iteration the source envelopes are transformed using regression coefficients and then aligned with target envelopes using DTW. After alignment LR coefficients are updated using the least squares algorithm.

2.3. Training the ANN

For voice conversion we use a feed-forward ANN with rectified linear units that implement the function $RL(x) = \max(0, x)$. The network performs mapping of the source to target envelope vectors (denoted as X and Y respectively) and consists of four layers as shown in figure 1.

Though parallel utterances contain the same aligned linguistic information the envelopes of source and target speakers cannot be perfectly matched because of pronunciation variations such as different intonations and excitation types. The problem is referred to as oversmoothing and can be partially solved in GMM-based approach by considering GV of the converted spectra [2].

Here in order to reduce oversmoothing we use variation of the speakers' state vectors $S_s = [f_{0,s}^{min}, f_{0,s}^{max}, u_s]$ and $S_t = [f_{0,t}^{min}, f_{0,t}^{max}, u_t]$ for source and target speaker respectively. The values of the vectors are calculated separately for each speaker using allowed minimum and maximum pitch values determined from speaker's pitch statistics:

$$f_{0,x}^{min}(n) = \begin{cases} (1 - u_x(n)), & f_{0,x}(n) < F_x^{min} \\ \dfrac{f_{0,x}(n) - F_x^{max}}{F_x^{min} - F_x^{max}}(1 - u_x(n)), & F_x^{min} \le f_{0,x}(n) \le F_x^{max} \\ 0, & f_{0,x}(n) > F_x^{max} \end{cases}$$

$$f_{0,x}^{max}(n) = \left(1 - f_{0,x}^{min}(n)\right)(1 - u_x(n))$$

where x stands for source or target speaker, F_x^{min} – minimum allowed pitch value, F_x^{max} – maximum allowed pitch value, $f_{0,x}(n)$ – current pitch value, $u_x(n)$ – current unvoiced flag that equals to 1 when frame n is unvoiced and 0 otherwise. Both S_s and S_t vectors contain normalized values in the range $[0,1]$.

In the conversion phase the state values related to the target speaker are calculated using converted pitch values and unvoiced flags of the source speaker (we assume that conversion does not change excitation type).

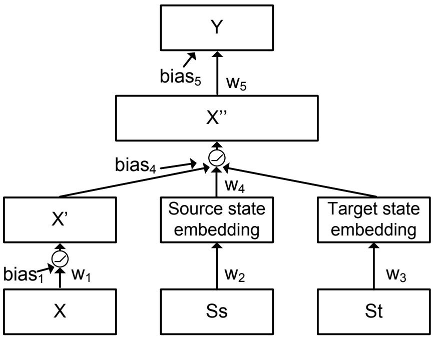


Figure 1: *Architecture of the proposed ANN*

The network represents the following mapping function:

$$Y = w_5 RL\left(w_4 \begin{bmatrix} RL(w_1 X + bias_1) \\ w_2 S_s \\ w_3 S_t \end{bmatrix} + bias_4\right) + bias_5$$

where square brackets denote vertical concatenation, w_{1-5} and $bias_{1-5}$ are weights and biases of the correspondent network connections.

The backpropagation algorithm is used to find the best parameters of the network minimizing the squared error between output vectors Y and actual vectors of the target speaker T:

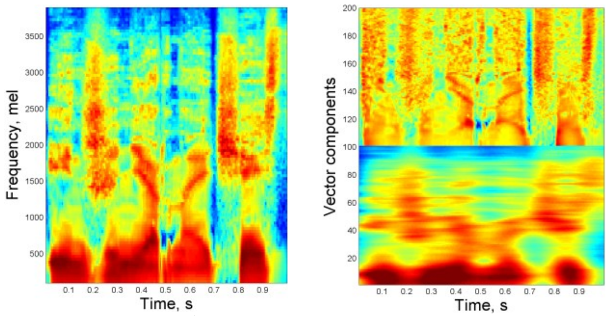
$$E = \sum (T - Y)^2$$

For backpropagation the gradient of $RL(x)$ is set to 0 when $x \le 0$ and 1 when $x > 0$ ignoring discontinuity at $x = 0$.

2.4. Improving ANN's performance

2.4.1. Incorporating static and dynamic features

A separate analysis frame does not contain enough information for performing an accurate envelope mapping. It is known that appending dynamic features sometimes significantly improves conversion quality [5]. We found that appending neighbouring frames, deltas or delta-deltas is not productive considering high computational cost. However we can get an evident enhancement by decomposing time series of envelopes into low-frequency and high-frequency parts using a low-pass filter with cut-off frequency 4-9Hz. The low frequency amplitude modulations capture speaker specific features while high frequency modulations contain phonetical and articulatory information [11]. The architecture of the ANN remains the same, however dimensionality of input vectors doubles as shown in figure 2.



source envelopes input vectors *X*

Figure 2: *Frequency decomposition of input envelopes*

2.4.2. Splitting the network and pretraining

A straightforward application of the discussed neural network requires many hidden neurons to achieve an acceptable conversion result. Assuming independence of the output envelope values the output vector *Y* can be split into *N* vectors of lesser dimensionality. That simplifies the mapping task reducing it to *N* independent tasks – figure 3. The overall number of parameters needed to fit the data can be much smaller and therefore training of the network can be performed much faster. Another advantage is capacity for parallel training using a multicore processing unit.

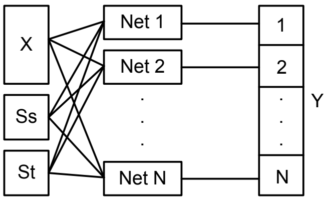


Figure 3: *Architecture of the split ANNs*

Initialization of the coefficients is made by random small values except the second hidden layer to output weight matrix w_5 which is initialized using target data. The target vectors are clustered using K-means algorithm and the number of clusters equals to the number of neurons in the layer. The columns of matrix w_5 are filled with centroids of the clusters. Such simple pretraining dramatically increases convergence speed and overall conversion accuracy. Further improvements can be made by using a sophisticated pretraining technique such as Restricted Boltzmann Machines [12].

2.4.3. Configuration of the network

Experimentally adjusting model parameters the following optimal network configuration (considering training time and conversion quality) has been found. The first hidden layer: number of neurons – 20, dimensionalities of state embeddings – 10; the second hidden layer: number of neurons – 20; dimensionality of output vectors – 5. Since target spectral envelopes are represented as 100-dimension vectors (for sample rate 44.1kHz) the number of nets *N*=20.

For 60 seconds of training data a MATLAB implementation of the ANN's learning algorithm takes 4 minutes to train on an Intel Core 2 Duo CPU (T6400 2.00 GHz using both cores).

2.5. Pitch mapping

Since it is hard to implement contextual pitch transformation in real-time we use the linear mapping function that has proved to be satisfactory for the task [5].

3. Real-time voice conversion

3.1. Inherent delay

The ANN discussed above operates on frame-by-frame basis and does not introduce any delays. However, frequency decomposition of envelope vectors requires low-pass filtering that implies the group delay of 100ms. Feature extraction requires additional 100ms including pitch and harmonic/noise parameters estimation. In waveform synthesis two filter banks are used (for harmonic and noise components) with subsequent signal unwrapping that needs 50ms delay. The overall inherent delay of the conversion system (constant time lag between source and processed signals) is 250ms.

3.2. Computational complexity

Features are extracted with 5 ms shift that requires execution of 200 forward-propagation algorithms per second. According to the parametric model the dimensionalities of the input and output envelope vectors depend on sample rates of the input and output signals as shown in table 1.

Table 1. *Dimensionalities of spectral envelope vectors for different sample rates*

Sample rate (kHz)	8	11.025	16	22.05	32	44.1
Input vectors dimensionality	108	124	144	162	182	200
Output vectors dimensionality	54	62	72	81	91	100

Computational complexity therefore is different for each sample rate combination. Table 2 shows number of multiplications per second needed for forward-propagation for different source/target sample rates. According to the table the mapping can be successfully done even on relatively slow modern CPUs.

Table 2. *Multiplications needed for spectral mapping using proposed ANN (millions per second)*

Source sample rate (kHz)	Target sample rate (kHz)					
	8	**11.025**	**16**	**22.05**	**32**	**44.1**
8	7.7	8.3	9.6	10.9	12.2	12.8
11.025	8.4	9.1	10.4	11.8	13.2	13.9
16	9.3	10.1	11.6	13.2	14.7	15.5
22.05	10.1	11.0	12.6	14.3	16.0	16.8
32	11.1	12.1	13.9	15.8	17.6	18.6
44.1	11.9	12.9	14.9	16.9	18.8	19.8

A C++ implementation of the whole real-time voice conversion system (including feature extraction and synthesis in 44.1 to 44.1 mode) has been tested on an Intel Core 2 Duo CPU (T6400 2.00 GHz using one core). The average CPU core usage is about 80%.

4. Experiments and results

4.1. Experiments setup

Taking into account the native language of the listeners participated in subjective evaluations a Russian speech database has been recorded. The database contains full-band utterances of 20 different speakers (10 males and 10 females) sampled at 44.1kHz. For each speaker there are 60 phrase sets for training and 4 phrase sets for conversion. The conversion system is evaluated only in 44.1 to 44.1 mode.

4.2. Compared mapping methods

The proposed method (labeled as 'ANN') is compared with general GMM and FW methods (labeled as 'GMM' and 'FW' respectively).

The implemented GMM-based mapping does not utilize GV because it requires a special adaptation for real-time. We used the conventional method based on expectation maximization algorithm with 32 mixture components as described in [2]. The FW implementation is made using bilinear warping function as described in [13].

Same analysis/synthesis routine is used for all the methods, same aligned vector sequences for training and same excitations and converted pitch for synthesis. Thus we eliminate other impacts on the conversion quality except spectral mapping function.

The conversion results are rated using objective and subjective evaluations. The performance of the GMM and FW-based mapping methods can differ from their original implementations because of the different parametric model of the signal.

4.3. Objective evaluations

The mel-cepstral distortion is used as an objective evaluation measure for mapping accuracy [2]:

$$\text{MCD} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{D}\left(\text{mc}_d^y - \text{mc}_d^t\right)^2}$$

where mc_d^y and mc_d^t are the d-th coefficients of the converted and target mel-cepstra respectively. The mean values of the MCD are listed in table 3.

Table 3. *Objective tests of the mapping methods (D=24)*

Number of training phrases	Mapping method		
	GMM	FW	ANN
10	4.8	5.6	4.3
30	4.6	5.5	4.1
60	4.5	5.5	3.9

The objective scores for GMM and FW-based methods are worse than for the proposed ANN method which is quite expected considering the complexities of the compared models. However these values do not reflect the real perceptual quality of the converted signals.

4.4. Subjective evaluations

A subjective mean opinion score (MOS) evaluations have been carried out. Twenty listeners were asked to rate (in 1-to-5 scale) similarity and perceptual quality (naturalness) between converted and target speech signals. The conversions have been made in each gender direction male-male, male-female, female-male, female-female (labeled as 'mm', 'mf', 'fm', 'ff' respectively) by each spectral mapping technique. Training for all methods in this experiment has been made on 26 training phrases (nearly 1 minute total) – the most likely case in the practical application. Some samples can be found on the following web-page "http://dsp.tut.su/Package.zip". The average results are summarized in figure 4.

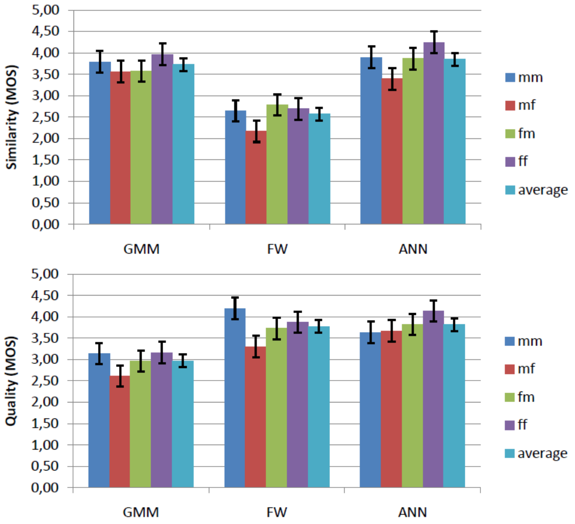


Figure 4: *Subjective evaluation of similarity and quality. Mean opinion scores for each conversion direction (at 95% confidence intervals)*

The performance of the GMM-based mapping is the worst in terms of subjective quality due to oversmoothing, however it is much better than FW-based mapping in terms of subjective similarity. The proposed ANN-based method achieves the highest average similarity value. The average quality score of the ANN-based method is very close to FW. The overall conversion quality of the proposed method is characterized by the listeners as 'near transparent'.

5. Conclusions

A real-time voice conversion technique based on ANN has been presented. The proposed architecture of the ANN utilizes ReLU and uses temporal speaker states in order to reduce the oversmoothing effect. The spectral mapping is scalable in the sense that it allows to process signals with different sample rates. The performance of the technique has been compared with GMM and FW-based mappings using objective and subjective measures.

6. Acknowledgements

The authors are grateful to the IT-Mobile company for their support[1].

[1] the conversion technique presented in the paper is used in the web services 'CloneVoice' and 'CloneAudioBook'. Both services are to become available for users on August 2013 at: http://clonevoice.com/en

7. References

[1] Y. Stylianou, O. Cappe, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. Speech Audio Process.,* Vol. 6, No. 2, pp. 131-142, 1998.

[2] T. Toda, A.W. Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Trans. Audio, Speech and Language Processing,* Vol. 15, No. 8, pp. 2222-2235, 2007.

[3] T. Toda, T. Muramatsu, and H. Banno. Implementation of computationally efficient real-time voice conversion. *Proc. INTERSPEECH*, Portland, USA, Sep. 2012.

[4] E. Godoy, O. Rosec, and T. Chonavel. Spectral envelope transformation using DFW and amplitude scaling for voice conversion with parallel or nonparallel corpora. *Proc. INTERSPEECH*, Florence, Italy, Aug. 2011.

[5] S. Desai, A.W. Black, B. Yegnanarayana, and B. Prahallad. Spectral mapping using artificial neural networks for voice conversion. *IEEE Trans. Audio, Speech and Language Processing,* Vol. 18, No. 5, pp. 954-964, 2010.

[6] D. Peng, X. Zhang, and J. Sun. Voice conversion based on GMM and artificial neural network. *Proc. ICCT*, pp.1121-1124, Nanjing, China, Nov. 2010.

[7] M. Narendranath, H.A. Murthy, S. Rajendran, and B. Yegnanarayana. Transformation of formants for voice conversion using artificial neural networks. *Speech Communication,* 16, pp. 207-216, 1995.

[8] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, G. Hinton. On Rectified Linear Units for Speech Processing. *Proc. ICASSP*, Vancouver, Canada, May 2013.

[9] E. Azarov, M. Vashkevich, and A. Petrovsky. Instantaneous pitch estimation based on RAPT framework. *Proc. EUSIPCO*, Bucharest, Romania, Aug. 2012.

[10] E. Azarov, and A. Petrovsky. Real-time voice conversion based on instantaneous harmonic parameters. *Proc. ICASSP*, Prague, Czech Republic, May 2011.

[11] S. Bacon, and D. Grantham. Modulation masking: effects of modulation frequency, depth, and phase. *Journal of acoustical society of America,* Vol. 85, pp. 2575-2580, 1989.

[12] V. Nair, and G.E. Hinton. Rectified linear units improve restricted Boltzmann machines. *Proc. ICML*, Haifa, Israel, June 2010.

[13] D. Erro, E. Navas, and I. Hernaez. Parametric voice conversion based on bilinear frequency warping plus amplitude scaling. *IEEE Trans. Audio, Speech and Language Processing,* Vol. 21, No. 3, pp. 556-566, 2013.