

A novel voice conversion system based on codebook mapping with phoneme-tied weighting

Zhi-Wei Shuang, Zi-Xiang Wang, Zhen-Hua Ling, Ren-Hua Wang

Iflytek Speech Lab of USTC, P.R.China
zwshuang@ustc.edu

Abstract

This paper presents a novel voice conversion system based on codebook mapping. A new phoneme-tied weighting strategy is proposed to reduce the smoothing effects in weighted sum of code books, while a new prosodic conversion method by decision tree is proposed to cope with the complex prosody of Chinese. STRAIGHT algorithm is used to decompose spectrum and excitation for separate modification. Listening tests prove the proposed methods can effectively convert speaker's individuality while maintaining high speech quality with a small amount of training data.

1. Introduction

With the development of corpus based TTS technique, the quality of synthesized speech has been improved a lot. But for the limits of capacity and costs of building corpus, we often only preserve few or only one speaker's speech data, so we can only synthesize speech with the timber of these specific speakers. However people often want to hear the voice they liked, so the technique of voice conversion attracts more and more researchers' interests.

To realize voice conversion, we usually need to decompose the speech signal into excitation signal and vocal tract filter, thus we can modify these two parts separately to recover the converted speech. Traditionally voice conversion is realized with formant synthesizer, and certain progress has been made. But the extraction of formant is very difficult, which often needs the manual mark and adjust of an expert. So it's almost impossible to be applied in corpus-based TTS system. And the voice quality of formant synthesizer is still not good enough to be accepted by users.

This paper proposed a novel voice conversion system based on STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weighted spectral contour). Speech signals are decomposed into impulse sequence and smoothed spectral contour. Spectral conversion and prosodic conversion are made separately.

For spectral conversion, there are two commonly used methods: by GMM and by codebook mapping. Here we use codebook mapping method because every codebook represents one segment of training data, which can preserve the information in the training data well, whereas in GMM methods, overall optimized conversion function for a group of data may lose information of some training data and cause smoothing effects of speakers' characteristics. And to solve the problem of dividing space into discrete curves in simple codebook mapping [1], we use weighted sum of code words to make the converted spectrum cover the target speaker's spectral space well. However, this method may introduce great smoothing effects in spectrum by summing quite different spectrums. To reduce smoothing effects, we propose

a new phoneme-tied weighting strategy which takes into accounts of the phoneme types and state types of code words in addition to the objective distance between spectral coefficients. The new strategy can reduce the smoothing effects greatly while maintaining high speech quality. What is more, since fewer weights need to be calculated in the new strategy, the calculation complexity decreases a lot, which is helpful for our strategy to be practically utilized. We also propose one novel prosodic conversion method based on decision tree to cope with complex prosody in Chinese. Subjective tests prove the effectiveness of the new models.

This paper will be arranged this way: Section 2 will briefly introduce our system's framework based on STRAIGHT coder. Section 3 will introduce our spectral conversion method in detail. Section 4 will introduce our prosodic conversion method based on decision tree in brief. Evaluation and discussion will be made in Section 5.

2. System Framework

2.1. Overview of Framework

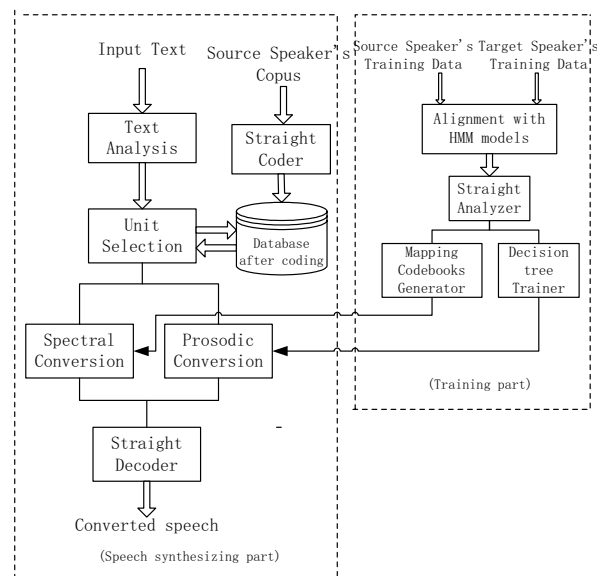


Figure1: Block of our TTS system with voice conversion

2.2. Introduction of STRAIGHT coder

STRAIGHT is an analysis-synthesis algorithm for speech signal. Its basic idea is to decompose the speech into excitation and vocal tract coefficients while it puts great emphasis on removing the effects of excitation from the spectrum. And the speech signal will be decomposed to independent spectral coefficients and excitation sequences.

Details can be referred to [2]. STRAIGHT algorithm is quite fit for voice conversion because it can reproduce high quality speech from the coefficients, and it can modify duration, F0 and spectral coefficients separately in large scale with little degradation to the quality.

In the original STRAIGHT algorithm, 1024 points FFT is used to present the spectrum with 1 ms frame shift. A large amount of data to restore the spectral coefficients is needed, about 10 times of the original speech data. So we need to make compression of spectral coefficients to make STRAIGHT be used in practical system. Here we use all-poles' model to present the spectrum first. 40 poles are used to make a precise model of the spectrum of speech data with 16K's samples per second. Then we transfer the poles to 40 LSF coefficients, which are easy to store and interpolate.

Also we only preserve one frame spectrum every 10 ms for several reasons. First, we find the quality of speech signal recovered from spectral data with frame shift of 10 ms is good enough.

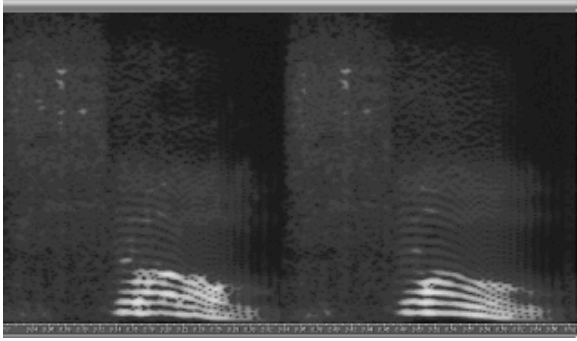


Figure 2: The converted spectrum of xiao2.

Secondly, we find the close spectrum may be discontinuous if we convert spectral data with too small frame shift, the close spectrum may be discontinuous if we make the conversion of frame separately, which will cause very uncomfortable jittering noise in the converted speech. In contrast, if we convert spectrum every 10ms and interpolate other frames, the jittering feelings can be successfully reduced and the quality improves a lot. The left spectrogram in Figure 2 is the converted spectrum with spectrum converted every 5 ms while other frames interloped. And the right one is the converted spectrum with spectrum converted every 10ms while other frames interloped. We can see clearly there are a lot of burrs in the left spectrum while these burrs are successfully smoothed in the right one. Finally, a larger frame shift can reduce the calculation complexity in conversion.

So 10ms' frame shift is fit for our voice conversion system. And the final code rate for speech data with 16K 's sampling rate is about 10 Kbps.

3. Spectral conversion by tied weighting

3.1. Basic method by weighted sum of codebooks

In codebook mapping method, first we should make accurate alignments between source and target speaker utterances. Using the alignments, source speaker acoustic characteristics are mapped to target speaker acoustic characteristics, spectral features extracted from corresponding segments of source

speaker and target speaker will form mapping code words. To convert the source speaker's voice, the simplest conversion method is to find the closest source speaker's code word and directly change it to the corresponding target speaker's code word. However the main problem of this method is to split the spectral space into a group of discrete curves, which will cause degradation in the quality of converted speech.

Here we use the method of weighted sum of spectral contours. For every frame of spectral contour presented by LSF coefficients, $w_k, k=1, \dots, P$, we first decompose it into a weighted sum of source speaker's code words, that is to find a group of v_i to make w_{sk}

$$w_{sk} = \sum_{i=1}^L v_i S_{ik}, k=1, \dots, P,$$

to be as close to $w_k, k=1, \dots, P$ as possible. Then the converted spectral contour is the weighted sum of the corresponding code words of target speaker, represented as

$$w_{tk} = \sum_{i=1}^L v_i T_{ik}, k=1, \dots, P,$$

It is very difficult to search for a optimal group of v_i , we use the method proposed by Levent M. Arslan to calculate the weights. Details can be referred to [3].

$$h_k = \frac{1}{\arg \min(|w_k - w_{k-1}|, |w_k - w_{k+1}|)} \quad (1)$$

$$k=1, \dots, P,$$

$$d_i = \sum_{k=1}^P h_k |w_k - S_{ik}|, i=1, \dots, L. \quad (2)$$

$$v_i = \frac{e^{-\gamma d_i}}{\sum_{i=1}^L e^{-\gamma d_i}} \quad (3)$$

3.2. Generation of initial mapping codebooks

First we select a group of phonetic balanced sentences from a text corpus. Then we ask both, the source speaker and the target speaker, to pronounce these sentences in a narrative mode. We use speaker-independent HMM models to segment all sentences. Every initial is segmented to 3 states, while every final is segmented to 5 states. Then we analyze all the speech data with STRAIGHT algorithm, speech data is transferred into sequence of spectral contours and F0. Spectral contour is restored with 40 LSF coefficients.

With the segmentation of state, we will extract the characteristic spectrum for each state. Here we directly extract the spectrum at the middle position as the characteristic spectrum of that state. We don't use the average spectral contour of that state as the characteristic spectral contour because we find the averaging operation cause the spectrum to be smoothed and lose the speaker's characteristics. And since the source speaker and the target speaker pronounce the same corpus, the states sequences of source speaker and target speaker are assumed to match one by one. These matched sequences are used as initial mapping codebooks.

To solve the problem of lack of code words, we split the mapping code words to 8 segments, with 5 LSF coefficients for each. Then for 40 LSF coefficients to be converted, we

also split them into 8 segments, and we calculate the weights for each segment separately. Thus, we can acquire quite good quality with only 4000 mapping code words.

3.3. A novel phoneme tied weighting method

3.3.1. Original weights calculation method's problem

The codebook mapping method is based on the well matched codebooks of source speaker and target speaker, especially those code words included in the same weighted sum. For one frame of source speaker's spectrum to be converted, the weights-calculating strategy will give large weights to those source speaker's code words close to the spectrum to be converted, which are also likely to be close to each other. If at the same time, the corresponding code words of target speaker are also close to each other, the weighted sum of these code words wouldn't cause much smoothing effects. Unfortunately, that is not the truth. Due to the errors in the segmentation and the variance of speaker's pronunciation, the corresponding target speaker's code words with large weights can be quite different. To include these code words in weighted sum will cause great muffling effects in converted spectrum, as shown in *the first spectrogram* in Figure 3.

3.3.2. Split codebooks to groups by the phoneme types

By statistics, we find mapping code words belonging to the same phoneme are quite close to each other. Obviously if we only select those mapping code words belong to the same phonemes, the similarity of code words included in weighting can be guaranteed. This method can greatly reduce the smoothing effects in spectrum. As shown in *the second spectrogram* in Figure 3, the formants is much more obvious than the first spectrogram which includes all mapping code words in weighted sum. By listening, the muffling effects are successfully removed.

But because there are one few mapping code words for one phoneme, especially when the training data is limited, one frame of spectrum to be converted may can not find close source speaker's code words at all. Thus degradations similar to those caused by error in speech coding may happen. From the second spectrogram, we can find two obvious fractures in the formant. These fractures can severely harm the quality of converted speech.

To solve this problem, we make some combinations of phonemes. We split all phonemes into several group based on previous literature and statistics, and only mapping code words belong to the same group can be included in one weighted sum.

Here, we propose the grouping strategy used in our system as below. However, modification of the strategy for different people may get better performance.

Table 1: Four groups for Finals

Group 1	"ii", "o", "e", "er", "ie", "u", "uo", "ve", "ao", "ou", "iao", "iou"
Group 2	"iii", "i", "v", "ai", "ei", "uai", "uei"
Group 3	"a", "ia", "ua"
Group 4	"an", "en", "ang", "eng", "ian", "in", "iang", "ing", "uan", "uen", "uang", "ueng", "ong", "van", "vn", "iong"

Table 2: Three groups for Initials

Group 1	"b", "p", "f", "d", "g", "k"
Group 2	"m", "n", "l", "r"
Group 3	"h", "j", "q", "x", "z", "c", "s", "zh", "ch", "sh"

3.3.3. Add the distances between phonemes to weights calculation

However, since the distance between different phonemes varies and the distance may also vary with state types, we need to consider this distance in our strategy to get better performance. Here, we split all the code words into different groups based on their phoneme types and state types. Then we calculate the distance between every two groups by averaging the distance between every two code words belong these two groups respectively. Thus we get a distance table between groups. We find this distance table matches the previous literature well. The distance table shows clearly, the distances between states of the same phoneme are quite small. Also, the distances among some well known close phonemes in Chinese, such as "j, q, x", "z, c, s" are very small. The average distance between groups will be added to the objective spectral distance calculated by Equation 2 to calculate the weight with Equation 3 since such average distance is more robust than objective spectral distance.

The final result after grouping of phonemes and the usage of distance table between phonemes is shown as *the third spectrogram* in Figure 3.

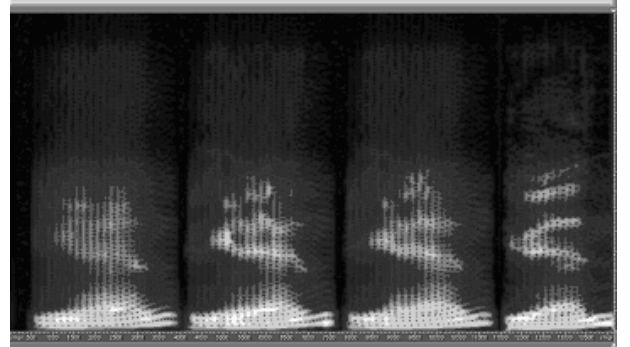


Figure 3 : Four spectrograms of Yuan2

We can see the new method effectively remove the fractures in formants in the second spectrogram while keeping the clearness of formants at the same time. Also, we can find the spectrogram of converted speech is quite close to *the fourth spectrogram*, which is calculated from the target speaker's real speech data.

4. Prosodic conversion by decision tree

The aim of prosodic conversion is to make the converted speech sound like the target speaker in prosody. And the most important prosodic feature for Chinese voice conversion is F0. Since the F0 is much more complex in Chinese, the method of pitch range conversion, which is often used in English, is not fit for Chinese. Here we use decision tree to make a better estimation of modification rate for F0 conversion.

First, we extract the F0 curves of each syllable from source speaker and target speaker's speech data with STRAIGHT algorithm. After smoothing the F0 curves, all the

curves are normalized to 10 points. Then for each syllable in the training text, we calculate the ratio curve between the target speaker's F0 curve and the source speaker's F0 curve. All the ratio curves are clustered to 5 clusters. Then we use the clusters' Nos. as the target variables while using the syllables' prosodic environment features as factor variables to train a decision tree [4]. Here, we cluster only 5 clusters to reduce the problem of data scarcity, to reduce the effects of noisy data.

We calculate the importance of features after training the decision tree. We find the most important three features are: the tone type of the syllable, the position in a breathing segment, the position in the sentence. And the importance of the tone type is much higher than the other two features. It may imply that different persons may have different patterns for each tone type. This opinion is similar to Professor Chilin Shih's opinion in STEM-ML [5], in which four F0 templates, with one for each tone, are used to model the prosody of Chinese.

Duration modification rate will be estimated by another decision tree. Modified prosody will be sent to the STRAIGHT coder to produce converted speech.

5. Subject evaluation and discussion

To get better evaluation of our voice conversion part, we take off the unit selection module in our frame work by selecting all the units from one natural sentence.

We ask 2 males (M1, M2) and 2 females (F1, F2) to pronounce the same 50 sentences in narrative mode. Then we make four groups of voice conversion between speakers: Female 1 to Male 1, Male 1 to Female 1, Female 2 to Female 1 and Male 2 to Male 1. The first 40 sentences are used to train the model, and the trained model is used to convert the last 10 sentences. The converted speech is compared with the corresponding sentences of the source speaker and the target speaker to get a final evaluation. 7 listeners with more than 2 years' experience in speech are asked to give evaluation.

The first test is a preference test (ABX test). Listeners are asked to indicate their preference for each pair. We make 3 sets of voice conversion: we use our spectral conversion part for all three sets while using pitch range conversion for set1, using our prosodic conversion model for set2 and using the target speaker's prosody for set3. The result is listed below.

Table 3: Preference test

Conversion Type	F1-M1	M1-F1	F2-F1	M2-M1
Set 1	97%	99%	89%	56%
Set 2	100%	99%	96%	78%
Set 3	100%	100%	98%	92%

The results of Set1 and Set2 indicate that our prosodic conversion model can work much better in the conversion between speakers with the same gender than traditional pitch range conversion. Also, the results of Set3 prove our spectral conversion method is effective in converting speaker's individuality. However, the high scores of conversion between different genders for all three sets show the need for a more precise evaluation.

So we ask listeners to make an opinion test. We use 5 grades: 5 means very similar to target speaker, 0 means very similar to source speaker and 3 means no clear preference can

be made. We also ask listeners to give a Mean Opinion Score (MOS) to the quality of speech for set2.

Table 4: Opinion test

Conversion Type	F1-M1	M1-F1	F2-F1	M2-M1
Set 1	3.5	3.3	4.3	3.2
Set 2	4	3.8	4.5	3.6
Set 3	4.7	4.5	4.8	4.3
MOS (Set2)	3.3	3.1	3.3	3.5

The low opinion scores for F1-M1 and M1-F1's conversion in Set1 show that although listeners can identify the converted voice's gender correctly, they don't agree the converted speech is close to the target. And obvious improvements in Set2 prove our prosody conversion method's superiority to pitch range conversion. For the MOS, all four groups of conversions are higher than 3, and M2-M1's conversion get 3.5. Also, in our tests the speech quality with a male target speaker is better than that with a female target speaker. Though not for sure, we think it might be because that modification at a high frequency can cause more severe degradation to the quality.

We notice conversion from M2 to M1 get much lower preference scores than other conversions. It is mainly because M2 have some special pronunciation habits, such as a constant nasal for all syllables. And in current framework the converted speech will inhabit such habits somehow. Research will be made to solve this problem in the future.

6. Summary

A novel voice conversion system based on codebook mapping is presented in this paper. This novel system is characterized by a new phoneme-tied weighting strategy to reduce the smoothing effects in weighted sum. The system is also characterized by a new prosodic conversion method by decision tree to cope with the complex prosody of Chinese. STRAIGHT algorithm is used to decompose the spectrum and excitation to make separate medication. The results of listening tests show that, the proposed methods enables the conversion of speaker individuality while maintaining high speech quality with only small amount of training data.

7. References

- [1] M.Abe, S.Nakamura, K.Shikano and H.Kuwabara, "Voice Conversion through vector quantization" *Proceedings of ICASSP 1988*, pp. 655-658.
- [2] Kawahara H., "Restructuring speech representations using a pitch-adaptive time frequency smoothing and a instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sound", *Speech Communication* 27 (1999), pp. 187-207
- [3] Levent M. Arslan, "Speaker Transformation Algorithm using Segmenta Codebooks (STASC)", *Speech Communication* 28 (1999), pp. 211-226.
- [4] Jiawei Han, Micheline Kamber, *Data mining concepts Techniques*, Mogan Kaufmann Publishers, 2001
- [5] Shih, C., Kochanski, G.P., "Chinese tone modeling with Stem-ML", *Proc. Internat. Conf. on Spoken Language Processing*, Beijing, China, vol 2 (2000), pp 67-70