A VOICE CONVERSION METHOD MAPPING SEGMENTED FRAMES WITH LINEAR MULTIVARIATE REGRESSION

HUNG-YAN GU, JIA-WEI CHANG

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology,
Taipei 106, Taiwan
E-MAIL: guhy@mail.ntust.edu.tw, m9815064@mail.ntust.edu.tw

Abstract:

In this paper, we study a different spectral mapping mechanism based on linear multivariate regression (LMR). Such LMR based spectral mapping methods are intended to alleviate the problem of spectral over-smoothing usually encountered by a GMM based method. First, we derive a solution formula to determine the best LMR mapping matrix. Then, for experimental evaluation, we record a parallel corpus, and adopt discrete cepstrum coefficients (DCC) as the spectral features. Next, we label and segment the recorded sentences into the speech units of syllable initials and finals. Hence, an LMR mapping matrix is trained for each syllable initial or final type. In terms of these LMR mapping matrices, we construct a voice conversion system. According to the measured average conversion errors, our system when using the mapping method, LMR_F, can indeed outperform a conventional GMM based voice conversion system. In addition, listening tests are conducted. The results show that the converted speech by our system is slightly better than that converted by a conventional GMM based system.

Keywords:

Voice conversion, Linear multivariate regression, Gaussian mixture model, Discrete cepstrum coefficients

1. Introduction

The purpose of voice conversion is to convert the speech of a source speaker into the speech of a target speaker. Recently, many researchers based on Gaussian mixture model (GMM) to study voice conversion [1], and tried to solve problems encountered. One of the typical problems encountered when GMM is adopted to map spectral coefficients is the phenomenon of over-smoothed converted spectral envelopes. One example is drawn in Figure 1. The dash-lined curve represents the spectral envelope of one frame uttered by the target speaker whereas the solid-lined curve represents the corresponding converted spectral envelope. Comparing these two curves, we see that the formants, F2, F4 and F6, on the solid-lined curve become broader, i.e. the depth from a peak to its left or right valley is decreased. When such over-smoothed spectral envelopes are used to synthesize speech signal, the resultant speech will be perceived as muffled and distorted.

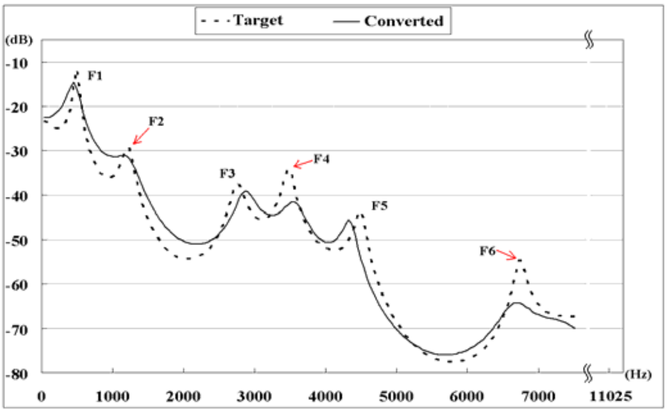


Figure 1. An over-smoothed converted spectral envelope

In this paper, we decide to study a different kind of spectral mapping method in the hope to prevent spectral over-smoothing from occurring. The mapping method adopted is LMR, and the criterion of least mean square (LMS) error is based to determine the optimal value for the mapping matrix. The concept of LMR is as follows. In the training stage, a mapping matrix, M, of size d×d is trained first with a parallel corpus. Here, d denotes the number of dimensions of a spectral feature vector. Then, in the conversion stage, a feature vector, S_k (size d×1), computed from the k-th frame of a source-speaker utterance will be converted to V_k with LMR. That is, let $V_k = M \bullet S_k$. We know that the idea of converting spectral envelope with LMR is not new and is already proposed in 1992 by Valbret, *et al*. [2]. Nevertheless, the solution method proposed to determine the matrix, M, is not complete in the regression constants. Therefore, we are motivated to study an analytic and complete solution method for the mapping matrix, M. The details of our derived formula are explained in Section 2.

In addition, we consider another problem mentioned in some previous works by other researchers [3], [4]. That is, the problem of one-to-many mapping will be encountered when voice is converted with the conventional GMM based methods. Such problem may result in that the converted spectral envelopes from some adjacent source frames become discontinuous, and such spectral discontinuities cause artifact sounds being synthesized. To alleviate the problem of one-to-many mapping, we thus decide to label and segment the recorded sentences into the speech units of syllable initials (e.g., /b/, /s/, /n/, *etc*) and finals (e.g., /a/, /ia/, /uai/, /ang/, *etc*). Next, the speech frames segmented to a same type of syllable initial or final are put together into a group corresponding to that type. Then, the frames collected in each group are used to train a dedicated LMR mapping matrix for that group's corresponding syllable initial or final. Nevertheless, in the conversion stage, how can we know to which syllable initial or final type an input frame belong? This is a problem to be solved as speech recognition. However, it needs not to be so serially treated as in speech recognition. An erroneously recognized but similar type of syllable initial or final may be tolerable for voice conversion. Also, we had studied a voice conversion method based on segmental GMM previously [3]. In that work, an algorithm for automatic selection of segmental GMM is proposed. That algorithm may be used to determine the initial or final types for a sequence of input source (source speaker) frames.

Another issue for voice conversion is the selection of spectral coefficients. Here, we continue to adopt discrete cepstrum coefficients (DCC) [5], [6] as the spectral features. The order of DCC is set to be 40. That is, 41 DCC, c_0, c_1, c_2, ..., c_{40}, are estimated from each frame. Among the 41 DCC, just 40 coefficients (c_0 eliminated) are used for spectral mapping. That is, the number of dimensions, d, is 40 here. After spectral mapping, the converted DCC are taken to compute their corresponding spectral envelope [5], [6]. Then, according to the spectral envelope and pitch frequency converted from a source frame, the harmonic and noise parameters' values for harmonic plus noise model (HNM) [6], [7] can be determined. Thereafter, those HNM parameters of successive frames are used to re-synthesize speech signal [6], [7], i.e. the converted speech signal.

2. LMR Mapping Matrix

In the training stage, speech segments belonging to a same segment type (e.g. /n/ is a syllable initial type and /ia/ is a final type) are put into a group. Then, each pair of parallel segments within the group are framed and aligned through dynamic time warping (DTW). Therefore, each source frame has an aligned target (target speaker) frame associated with it. Here, let S_1, S_2, ..., S_N, be the sequence of DCC vectors computed from the source frames. After DTW, another sequence of DCC vectors, T_1, T_2, ..., T_N, can be computed from the target frames aligned to the source frames. For convenience of derivation, let $S = [S_1, S_2, ..., S_N]$, i.e. let S be a d×N matrix consisted of N columns of source DCC vectors. Similarly, let $T = [T_1, T_2, ..., T_N]$, i.e. T is consisted of N columns of target DCC vectors. Ideally, we intend to find an LMR mapping matrix, M, of size d×d, in order that the relation of equality,

$$M \bullet S = T\,, \tag{1}$$

is held.

In practice, N is usually much larger than d. Therefore, an ideal mapping matrix, M, will not exist. That is, some error will be induced when a source DCC vector, S_k, is to be mapped to T_k through M•S_k. Here, let E be the error matrix, of size d×N, whose definition is

$$E = M \bullet S \; - \; T\,. \tag{2}$$

The goal, to find an optimal mapping matrix, M, is equivalent to minimize the absolute values of all the elements of E. Notice that the number of elements in E is d×N, which is much larger than d×d, the number of elements in M. Therefore, the criterion of LMS is adopted here, and a matrix, $\mathcal{E}$, consisted of squared errors is computed first. The definition of $\mathcal{E}$ is

$$\mathcal{E} = E \cdot E^{\mathrm{t}} = (M \cdot S - T)(M \cdot S - T)^{\mathrm{t}}, \quad \mathrm{t}: \text{transpose.} \tag{3}$$

Then, the trace of $\mathcal{E}$, i.e. $\mathrm{tr}(\mathcal{E}) = \mathcal{E}_{1,1} + \mathcal{E}_{2,2} + ... + \mathcal{E}_{d,d}$, is partially differentiated with M, and the result of the partial differentiation is set to be a zero matrix [5, 6]. The corresponding formula is

$$\frac{\partial\left(\mathrm{tr}(\mathcal{E})\right)}{\partial M} = 2(M \cdot S - T) \cdot S^{\mathrm{t}} = 0 \quad . \tag{4}$$

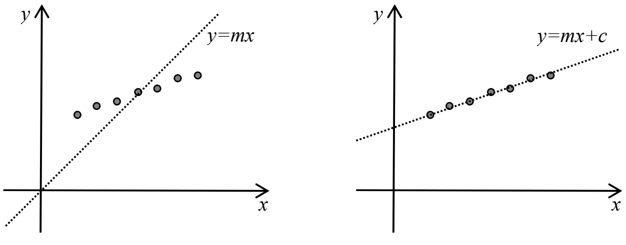
In Equation (4), the matrix-form notation, $\partial\left(\mathrm{tr}(\mathcal{E})\right) \;/\; \partial M$, is actually meant to denote $\partial\left(\mathrm{tr}(\mathcal{E})\right) \;/\; \partial M_{i,j}$, j=1, 2, ..., d, i=1, 2, ..., d, in a compact form. After rearranging Equation (4), the formulas below are derived. That is, an analytic solution for the mapping matrix, M, is obtained.

$$M \cdot S \cdot S^{\mathrm{t}} = T \cdot S^{\mathrm{t}}\,, \tag{5}$$

$$M = T \cdot S^{\mathrm{t}} \cdot (S \cdot S^{\mathrm{t}})^{-1} \quad . \tag{6}$$

Now, in terms of Equation (6), a local optimal solution for M can be obtained. The matrix M obtained is just a local optimal. Consider the example of univariate linear regression drawn in Figure 2(a). The regression line is constrained to pass the origin point, which will inevitably induce larger regression errors as compared with the errors induced in

Figure 2(b). Note that Figure 2(a) can be viewed as a visual example for the mapping matrix *M* given in Equation (1). That is, the regression line is constrained to pass the origin point, and thus induces larger regression errors.



(a) $y = m \cdot x$ (b) $y = m \cdot x + c$

Figure 2. Examples of univariate linear regressions

In this paper, we study to eliminate the constraint, passing the origin point. According to the example, Figure 2(b), it is seen that a constant term must be added to each dimension of the target vector (i.e. y in Figure 2) in order to eliminate the constraint. Hence, the method proposed here is as follows. First, the definitions of the three matrices, *M*, *S*, and *T*, are extended to $\tilde{M}$, $\tilde{S}$, and $\tilde{T}$ in the manner as

$$\tilde{M} = \begin{bmatrix} \multicolumn{1}{c}{\Large M} & \begin{matrix} M_{1,d+1} \\ M_{2,d+1} \\ \vdots \\ M_{d,d+1} \end{matrix} \\ 0, 0, ..., 0, & 1 \end{bmatrix}, \quad \tilde{S} = \begin{bmatrix} S_1 & S_2 & \ldots & S_N \\ 1, & 1, & \ldots & 1 \end{bmatrix}, \quad \tilde{T} = \begin{bmatrix} T_1 & T_2 & \ldots & T_N \\ 1, & 1, & \ldots & 1 \end{bmatrix}. \qquad (7)$$

In Equation (7), the matrix, $\tilde{M}$, is formed by placing the matrix, *M*, to the upper left corner, adding a (*d*+1)-th row of constant values (all 0 except the last 1), and adding a (*d*+1)-th column of variable elements. As to the matrices, $\tilde{S}$, $\tilde{T}$, they are extended from *S* and *T* by adding the (*d*+1)-th row of constant values (all 1). Then, the three matrices, $\tilde{M}$, $\tilde{S}$, $\tilde{T}$, are taken to replace the matrices, *M*, *S*, and *T* in Equation (6) to solve the value of the optimal mapping matrix, $\tilde{M}$. Consequently, the regression error induced when applying the mapping, $\tilde{M} \bullet \tilde{S}$, will be reduced.

3. System Implementation – Training Stage

For the voice conversion system built here, its processing flow for the training stage is as that shown in Figure 3. First, we invited two male speakers, denoted MSA and MSB, and two female speakers, denoted FSA and FSB, to record 375 Mandarin parallel sentences in a soundproof room. The number of syllables recorded is totally 2,926 for each speaker, and the sampling rate is 22,050 Hz. Among the four speakers, four speaker-pairs are associated to conduct voice conversion experiments, i.e. (MSA, MSB), (MSA, FSA), (FSA, MSA), and (FSA, FSB). In each pair, the former is the source speaker whereas the latter is the target speaker.

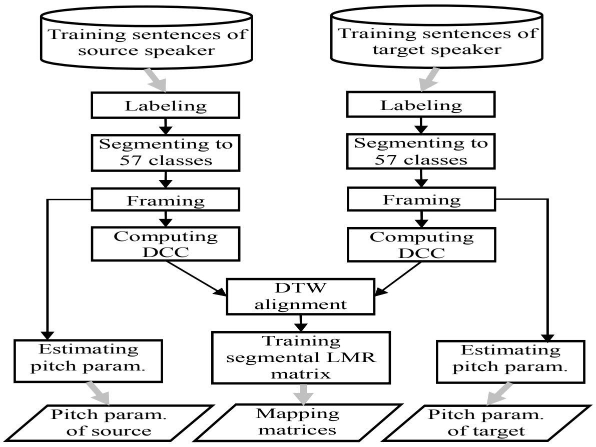


Figure 3. The processing flow for the training stage

3.1. Labeling and segmentation

The first 350 sentences of the recorded sentences are taken to train the conversion models whereas the 25 left sentences are used to test the conversion methods. First, these sentences are labeled by forced alignment with HTK (HMM tool kit). The boundary positions of each speech segment can thus be roughly obtained. Here, a speech segment is either a syllable initial or a syllable final. In Mandarin, a syllable initial is a consonant, and a syllable final is a vowel or diphthong with possibly an ending nasal phoneme. Since the boundary positions given by HTK are not accurate enough, we have to check the boundaries of each segment and correct them manually with the software package, Wavesurfer.

After labeling, the speech segments of the training sentences are grouped according to their labels. Here, we group these segments to 57 groups, including 21 syllable initial groups and 36 final groups.

3.2. DCC computation and DTW alignment

The length of a frame is 512 sample points (23.2 ms) and the frame shift is 128 points (5.8 ms). For each frame, 41 DCC coefficients are computed with a program module developed previously [6].

Because the speaking rates of the source and target speakers may be inconsistent, the sequence of frames sliced from a source segment must be aligned with the frame sequence sliced from its corresponding target segment. This alignment is done through DTW as usual.

3.3. LMR matrix computation

An LMR matrix was trained for each of the 57 groups. Since a group has several parallel segments collected, we have to separately align each pair of source and target segments through DTW. Then, the *j*-th target segment's aligned frame sequence is concatenated to the aligned frame sequence of the (*j*-1)-th target segment, and so on. Next, DCC vectors can be computed from the concatenated frame sequence and form the matrix, *T*, used in Equation (1). On the other hand, the source segments' frame sequences are concatenated directly, and DCC vectors are computed from the concatenated frame sequence and form the matrix, *S*, used in Equation (1). In terms of the two matrices, *S* and *T*, the basic LMR mapping matrix, *M*, can then be computed by applying Equation (6). In addition, after extending *S* and *T* to $\tilde{S}$ and $\tilde{T}$ as in Equation (7), we can compute the full LMR mapping matrix, $\tilde{M}$, according to Equation (6), too.

3.4. Pitch parameters

For each frame, the zero-crossing rate (ZCR) is computed first to determine if it has high ZCR value and is thus unvoiced. If a frame does not have high ZCR value, a pitch detection method based on autocorrelation function and AMDF [8] is then used to compute the pitch frequency of the frame. A frame may still be decided to be unvoiced if it does not pass the periodicity checking rules. For each speaker, the pitch frequencies of his voiced frames are computed first. Then, the average value and standard deviation of these pitch frequencies are calculated in logarithmic scale. These two values, average and standard deviation, are the pitch parameters adopted here for a speaker.

4. System Implementation – Conversion Stage

The processing flow of our system in the conversion stage is shown in Figure 4. After a spoken sentence of the source speaker is inputted, it is first sliced into a sequence of frames. The frame length and shift are same as those mentioned in Section 3.2. Then, in the left flow of Figure 4, the pitch frequency of each frame is detected. If a frame is detected to be unvoiced, the three gray colored blocks in Figure 4 will be skipped directly. That is, the pitch frequency of the frame is not defined and need not be adjusted, and the spectral parameters, DCC, will not be converted. On the other hand, if a frame is decided to be voiced, the "pitch adjusting" block will be executed, and the formula for adjusting pitch is as Equation (8),

$$q_t = \mu^{(y)} + \frac{\sigma^{(y)}}{\sigma^{(x)}}(p_t - \mu^{(x)}) \ , \tag{8}$$

where p_t is the detected frequency of a source frame, $\mu^{(x)}$ and $\sigma^{(x)}$ denote the average and standard deviation of the source speaker's pitch frequency, and $\mu^{(y)}$ and $\sigma^{(y)}$ denote those of the target speaker's pitch frequency.

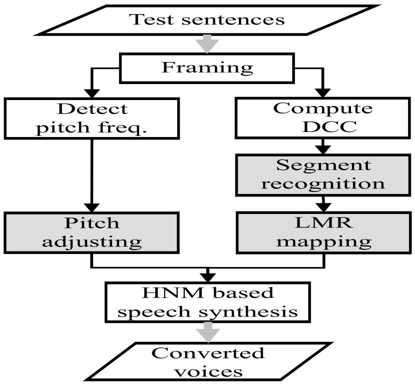


Figure 4. The processing flow for the conversion stage

4.1. Speech segment recognition

Notice that the focus of this paper is to study the conversion capability of the spectral mapping method based on LMR. Hence, the function of the block, "Segment recognition", is currently replaced with the label file corresponding to the input sentence. That is, the segment boundaries and phonetic labels for the speech segments comprising an input sentence are read from its corresponding label file directly.

In the future, we may implement the function of segmentation recognition with HTK to treat the case that the sentence is on-line uttered. As another choice, we may apply the algorithm proposed in our previous work [3] to determine segment boundaries and types (syllable initials and finals) automatically.

4.2. HNM based speech synthesis

In HNM, the spectrum of a voiced frame is split into two parts, i.e. lower frequency harmonic part and higher frequency noise part. The boundary frequency between the two parts is termed the maximum voiced frequency (MVF) [7]. To simplify the processing of speech signal synthesis, the MVF values of voiced frames are all fixed to 6,000 Hz in this study.

For a voiced frame, its DCC coefficients will be mapped by the block, LMR mapping, in Figure 4. This block is however bypassed for an unvoiced frame. Then, the DCC coefficients are inversely transformed to obtain a curve of spectral envelope. According to the spectral envelopes of successive frames, speech signal is synthesized with an HNM based scheme. That is, harmonic signal and noise signal are separately synthesized and then added to obtain the final speech signal. The synthesis processing with HNM will not be detailed here because the details can be found from our previous works [3, 6].

5. Experimental Evaluation

In Section 2, two LMR based mapping methods are proposed. The first method is to apply the matrix, *M*, defined in Equation (1) to map the DCC coefficients of a source frame. This method is termed the basic LMR mapping and is denoted as LMR_B. In contrast, the second method is termed the full LMR mapping and is denoted as LMR_F. In the method, LMR_F, the mapping matrix, $\tilde{M}$, defined in Equation (7) is used instead.

5.1. Measuring conversion error

Notice that 375 sentences are recorded from each speaker and only the first 350 sentences are used to train the LMR mapping matrices. Therefore, the 25 remaining sentences (totally 209 syllables) are used here for outside testing whereas the first 350 sentences are used for inside testing. In addition, for the purpose of comparison, we also used the fist 350 sentences to train a conventional GMM consisting of 128 Gaussian probability distributions [1]. The GMM based mapping method is denoted as GMM_128.

Let $R = R_1, R_2, \cdots, R_N$ be the sequence of converted DCC vectors, and $T = T_1, T_2, \cdots, T_N$ be the corresponding sequence of target DCC vectors. To measure the error induced by a conversion method, we use the formula,

$$D_{avg} = \frac{1}{N} \sum_{1 \le k \le N} dist(R_k, T_k) , \qquad (9)$$

to compute the average conversion error between the two sequences, *R* and *T*. In Equation (9), the function, *dist*( ), calculates the geometric distance between the two vectors, R_k and T_k. In terms of Equation (9), we measure the average conversion errors induced by the three voice conversion methods, LMR_B, LMR_F, and GMM_128. In addition, Equation (9) is applied four times each for one of the four speaker pairs, (MSA, MSB), (MSA, FSA), (FSA, MSA), and (FSA, FSB). Then, a gross average, AVG, is computed. In details, the values of the average conversion errors for the three methods are listed in Table 1.

From the first and second columns of Table 1, it can be seen that the full LMR based method, LMR_F, is as expected better than the basic LMR based method, LMR_B, in the conversion error induced. The reductions in conversion error are 1.6% and 1.7% respectively for the inside and outside tests. In addition, from the second and third columns of Table 1, it is found that the method, LMR_F, studied here can outperform the conventional GMM based method, GMM_128, in both inside and outside tests. The reductions in conversion error are 7.1% and 1.5% respectively for the inside and outside tests. Therefore, the method, LMR_F, is expected to obtain better timbre similarity and speech quality than the conventional GMM based conversion method if the source speech signal is segmented first before it is converted.

TABLE 1. AVERAGE ERRORS MEASURED WITH THE CONVERSION METHODS, LMR_B, LMR_F, AND GMM_128

Conversion errors		LMR_B	LMR_F	GMM (128 mix.)
Inside tests	MSA=> MSB	0.4890	0.4794	0.5058
	MSA=> FSA	0.4782	0.4705	0.5012
	FSA=> MSA	0.4967	0.4881	0.5412
	FSA => FSB	0.5514	0.5443	0.5853
	AVG	**0.5038**	**0.4956**	**0.5334**
Outside tests	MSA=> MSB	0.5467	0.5331	0.5346
	MSA=> FSA	0.5174	0.5106	0.5147
	FSA => MSA	0.5388	0.5307	0.5551
	FSA => FSB	0.5867	0.5782	0.5806
	AVG	**0.5474**	**0.5382**	**0.5463**

5.2. Subjective speech quality tests

Two sentences not used in training the models are taken to prepare four converted voice files for listing tests. These four files are denoted as X1, X2, Y1, and Y2. Among the four files, X1 and X2 are obtained with the voice conversion method, GMM_128 whereas Y1 and Y2 are obtained with the voice conversion method, LMR_F. The number "1" in X1 and Y1 means that the voice conversion is done between the speaker pair, (MSA, MSB). Similarly, the number "2" in X2 and Y2 means that the voice conversion is done between the speaker pair, (MSA, FSA). These four speech files can be accessed at http://guhy.csie.ntust.edu.tw/VCLMR/LMR.html .

In terms of the four converted speech files, two runs of listening tests are conducted to compare their speech quality. In the first run, X1 and Y1 are played in a random order to the listener, and the listener is requested to give a score. In the second run, the other two files, X2 and Y2, are played in a random order to the listener, and the listener is requested again to give a score. Here, 15 university students are invited to take part in the two runs of listening tests. Most of them are not familiar with the research field of voice conversion. As to the scores that a listener may give are -2, -1, 0, 1, and 2. The score, 2 (-2), means the quality of the latter played file is apparently better (worse) than the former played. The score, 1 (-1), means the quality of the latter played file is slightly better (worse) than the former played. On the other hand, the score, 0, means the quality of the two played files cannot be distinguished.

After listening tests, the scores given by the listeners are rearranged, and their averages and standard deviations are calculated for the two runs respectively. The results are listed in Table 2. According to the average scores, 0.867 and 0.467, it can be said that the converted voice by LMR_F will have slightly better speech quality than the converted voice by GMM_128. Therefore, the conversion method, LMR_F, can not only reduce conversion error but also promote speech quality as compared with the method, GMM_128.

TABLE 2. AVERAGE SCORES OBTAINED FROM THE LISTENING TESTS

Average (std. dev.)	GMM_128 vs. LMR_F
X1 vs. Y1	0.867 (0.640)
X2 vs. Y2	0.467 (0.704)

6. Conclusion

In this paper, a spectral mapping method based on LMR is studied, and a formula for determining the best value of an LMR matrix is derived. Then, we build a voice conversion system with the LMR based spectral mapping method, and evaluate the performance of this system experimentally. For building this system, parallel speech corpus is recorded, spectral coefficients, DCC, are adopted, and each recorded sentence is labeled and segmented into syllable initials and finals.

According to the average conversion errors measured, it is found that the errors induced by our method, LMR_F, are less than those induced by the conventional GMM based method, GMM_128. The reductions of conversion error are 7.1% and 1.5% respectively for the inside and outside tests. In addition, subjective listening tests are conducted to compare the speech quality of the converted voice files by the two methods, LMR_F and GMM_128. The results of the listening tests show that our method, LMR_F, can indeed obtain better speech quality than the method, GMM_128.

References

[1] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion”, IEEE trans. Speech and Audio Processing, Vol. 6, No. 2, pp. 131-142, 1998.

[2] H. Valbret, E. Moulines, and J. P. Tubach, “Voice transformation using PSOLA technique”, Speech Communication, Vol. 11, No. 2-3, pp. 175-187, 1992.

[3] H. Y. Gu, and S. F. Tsai, “An improved voice conversion method using segmental GMMs and automatic GMM selection”, Int. Congress on Image and Signal Processing, pp. 2395-2399, Shanghai, China, 2011.

[4] E. Godoy, O. Rosec, and T. Chonavel, “Alleviating the one-to-many mapping problem in voice conversion with context-dependent modeling”, Proc. INTERSPEECH, pp. 1627-1630, Brighton, UK, 2009.

[5] O. Cappé, and E. Moulines, “Regularization techniques for discrete cepstrum estimation”, IEEE Signal Processing Letters, Vol. 3, No. 4, pp. 100-102, 1996.

[6] H. Y. Gu, and S. F. Tsai, “A discrete-cepstrum based spectrum-envelope estimation scheme and its example application of voice transformation”, International Journal of Computational Linguistics and Chinese Language Processing, Vol. 14, No. 4, pp. 363-382, 2009.

[7] Y. Stylianou, Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification, Ph.D. thesis, Ecole Nationale Supèrieure des Télécommunications, Paris, France, 1996.

[8] H. Y. Kim, et al., “Pitch detection with average magnitude difference function using adaptive threshold algorithm for estimating shimmer and jitter”, 20-th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, Hong Kong, China, 1998.