SINGING-VOICE SYNTHESIS USING DEMI-SYLLABLE UNIT SELECTION

HUNG-YAN GU, JIA-KANG HE

Department of CSIE, National Taiwan University of Science and Technology, Taipei, Taiwan
E-MAIL: guhy@csie.ntust.edu.tw, m10115078@mail.ntust.edu.tw

Abstract:

In this study, an algorithm having a nice dynamic-programming structure is proposed for unit selection. This algorithm considers the costs of pitch and duration transformations, and the costs of contextual and spectral discontinuities. Here, the voice unit, demi-syllable, is adopted. In the training phase, each demi-syllable unit is analyzed to obtain a sequence of DCC (discrete cepstral coefficient) vectors. Then, in the synthesis phase, the pitch and duration of a syllable can be adjusted. In addition, the singing voice signals are synthesized with HNM (harmonic plus noise model) model. To evaluate the performance of our unit selection algorithm, we have conducted two listening tests. One test is to evaluate the spectral fluency (continuity), and the other test is to evaluate the synthesized songs' quality. The results of both tests show that our algorithm can improve a synthesized song's fluency level and quality noticeably.

Keywords:

Singing voice synthesis; Unit selection; Demi syllable; Discrete cepstral coefficient; Harmonic plus noise model

1. Introduction

Under the situation, using only a very limited set of syllable utterances to analyze the parameters of the HNM signal models, we had developed a Mandarin singing voice synthesis method based on HNM [1]. The synthesized song signals are very clear, i.e. of high signal-quality. Nevertheless, the synthesized song when heard is felt lacking of singing resonance. This problem we think is due to the recorded utterances which are uttered in speech style. Therefore, in this study, we record several songs sung by a real singer and use these songs to analyze the required features, e.g. spectral envelope parameters.

As discussed in other researchers' works [2, 3], singing synthesis systems are classified to model based and concatenative. An example of the model based systems is Sinsy [4], which is based on hidden Markov model (HMM). Recently, we had tried the HMM based tool kit, HTS [5], for our recorded songs to train HMM, and then use HTS synthesis engine to synthesize singing voice with the trained HMM. However, the synthesized songs are not satisfactory in general. First, the pitch contours of many synthesized syllables are of incorrect pitch heights or strange contours. These may be due to pitch detection errors in analyzing the recorded songs. Secondly, many syllables of the synthesized songs are perceived as muffled. The muffled voices we think is due to that many frames of generated spectral coefficients suffer in over-smoothed spectral envelopes.

In this study, we intend to prevent over smoothed spectral envelopes from being generated. Hence, we decide to adopt the approach of unit selection instead of HMM based spectral modeling. Conventionally, unit selection usually imply that the pitch and duration of a selected unit would be modified by a time-domain method, e.g. PSOLA [6]. In this study, however, we will modify a unit's acoustic characteristics (pitch and duration) with a frequency domain method, actually an HNM based method [1]. Our decision is based on that singing expressions, e.g. vibrato and portamento, are more convenient to present with a frequency domain method.

For the choice of unit size, we decide to take the size, demi-syllable. Apparently, the unit size, syllable, is too large because the number of different combinations of the relevant factors, syllable id, pitch id, duration class, and left and right context classes, will be very large. On the other hand, the unit size, diphone, is more difficult than demi-syllable to determine the boundary point between adjacent units. Therefore, we think the unit size, demi-syllable, is very appropriate for a syllable prominent language, e.g. Mandarin.

Besides unit selection and signal synthesis method, another important issue is how to synthesize a song as natural as sung by a real singer. We think the most relevant factor to naturalness is the singing expression of vibrato. In the past, we had studied ANN based models for generating vibrato parameters [7]. Such models are very effective in generating natural vibrato expressions. Accordingly, in this study, we just focus on the issue, developing an effective unit selection algorithm for demi-syllables.

2. System structure

A singing synthesis system is built in this study. We

implement this system in two stages, i.e. preparation and synthesis stages. The works done in the preparation stage is described in Section 2.1 whereas the processing steps for the synthesis stages are described in Section 2.2.

2.1. Preparation stage

In preparation stage, the recorded songs are segmented into phrases, and each phrase is labeled with syllable lyrics and syllable-boundary points. Then, each phrase is sliced into a sequence of frames, and each frame is analyzed for its pitch and spectral coefficients. The spectral coefficients adopted here are discrete cepstral coefficients (DCC) [8]. Next, each syllable extracted from a phrase is split into a left half-syllable (LHS) unit and a right half-syllable (RHS) unit. Such LHS and RHS units are the basic voice units adopted for unit selection.

Corpus and labeling

We invite a female singer to sing 44 Mandarin songs in a soundproof room. The sampling rate is set to 22,050 Hz. These songs consist of 5,882 syllables in total. After recording, these songs are manually segmented into phrases. Then, these phrases are automatically labeled with the software package, HTK. Since most of the boundary points are incorrectly labeled, we have to manually correct the wrongly labeled boundary points. Additionally, for each syllable, we add a boundary point between the initial consonant and the final vowel group.

Pitch detection and DCC calculation

Each frame of a syllable is analyzed to obtain its pitch value and DCC coefficients. For pitch detection, a method that combines autocorrelation and AMDF is adopted. After the frames of a syllable are pitch detected, the obtained pitch values are averaged in logarithmic scale to calculate an average pitch for this syllable. As to the representation of a frame's spectral envelope, we use DCC coefficients. For analyzing DCC, we use the program modules developed in the previous work [9].

Syllable splitting and context label

The voice unit is demi-syllable here. Each syllable extracted from a recorded song is split into an LHS unit and a RHS unit. As to determine the splitting point, we have ever trained an HMM from a syllable's frame sequence, and then select the frame that is occupied by the middle state of the HMM. Nevertheless, the selected frame may be very close to the beginning or ending of the nucleus vowels' duration in some syllables, i.e. very nonuniform splitting. Therefore, we decide to directly put the splitting point to the vowel's half duration for a syllable of a single nucleus vowel. For a syllable of several nucleus vowels, we directly put the splitting point to one third of the vowels' duration.

For Mandarin, the number of different LHS units is only 356 whereas the number of different RHS units is as less as 36. However, to synthesize a fluent song with spectral continuity, we have to consider contextual continuity. Therefore, we also mark each unit's left and right contexts besides the current unit's label. For example, the label tuple, <mai+, L(u), R(d)>, indicates that the current unit, labeled /mai+/, is the LHS of a syllable, /mai/, the left context is L(u) which means the tail phoneme of the previous syllable belongs to the phone class, /u/, and the right context is R(d) which means the leading phoneme of the next syllable belongs to the phone class, /d/. In contrast to /mai+/, the complement RHS unit would be labeled as <+ai, L(u), R(d)>.

2.2. Synthesis stage

For synthesis stage, the processing flow is drawn in Figure 1. The processing steps include (a) Input notes and lyrics, (b) Select demi-syllable units, (c) Generate pitch contours of vibrato expression, (d) Synthesize each syllable's signal samples with HNM, and (e) Concatenate syllable signals.

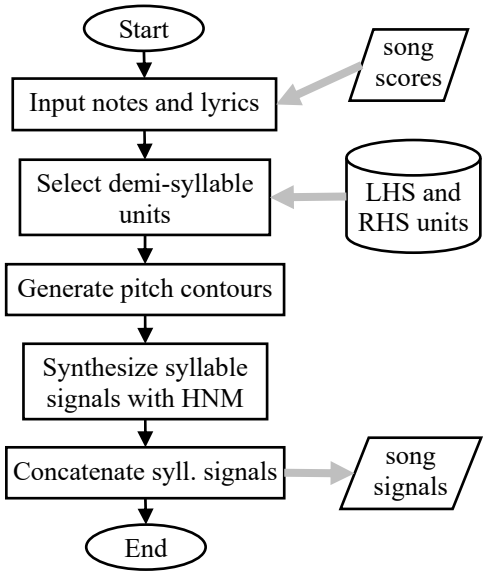

Figure 1. Chief processing flow for synthesis stage

Input notes and lyrics

The first line of a score file contains the data, song name and tempo. Then, each of the following lines contains the items, lyric syllable (e.g. /pau/), tone symbol (e.g. C4) or tone symbols (e.g. C4-E4) for portamento, and number of beats or numbers of beats (e.g. 2-2).

Select demi-syllable units

In this study, we have designed a dynamic programming based algorithm to execute unit selection. The details of this algorithm will be present in Section 3. For each lyric syllable, an LHS unit and a RHS unit would be selected from the deposit. Then, the two units' corresponding DCC files are concatenate to form the lyric syllable's DCC file.

Generate pitch contours

The pitch contour of a syllable should be generated with vibrato expression in order for natural singing voice to be synthesized. Here, "vibrato" means both local vibration and global trend within a pitch contour. We had studied an effective method, based on ANN, for generating vibrato parameters, i.e. intonation, and vibrato extent and rate [7]. Therefore, the program modules developed previously is directly applied here to generate pitch contours.

Synthesize syllable signals with HNM

After unit selection, each lyric syllable has its corresponding sequence of DCC vectors. Hence, the spectral envelope of a frame can be derived from its DCC vector [9]. Next, the f0 value generated for a fame (as a point in the pitch contour) can be used to determine each harmonic's frequency and amplitude (guided by the spectral envelope). According to the harmonics' frequencies and amplitudes and the spectral envelope of each frame, the signal model, HNM, can be used to synthesize a singing syllable's signal waveform. The details of HNM based signal synthesis are referred to the previous works [1, 9].

Concatenate syllable signals

By concatenating the synthesized signals for a sequence of lyric syllables, a synthesized song can be obtained. Nevertheless, such concatenation is not trivial in practice. For the synthesized song to have correct tempo, a syllable that has initial consonant must be placed ahead its corresponding note's start time. In more precise, we should align the vowel part of a syllable with the corresponding note's start time. Hence, overlap and add is performed for some adjacent syllables' signals.

3. Unit selection

The factors that affect the synthesis of a singing syllable include pitch, duration, LHS, RHS, and left and right context classes. The number of different combinations of these factors is very huge. Nevertheless, the number of collected LHS and RHS units is relatively small. Therefore, for practical implementation, cost functions are defined, and then unit selection is performed to minimize the accumulated cost in order for obtaining a best unit sequence.

3.1. Unit selection algorithm

Several types of costs are discussed in the work by M. Umbert, *et al*. [10]. Here, we adopt some of the cost types, i.e. transformation cost, continuity cost, and concatenation cost. Using these costs, we develop a dynamic programming based unit selection algorithm. Let SF(t, m) denote the m-th LHS candidate unit at time t (i.e. t-th syllable), SB(t, n) denote the n-th RHS candidate unit at time t, and Nm and Nn denote the numbers of candidate units for LHS and RHS, respectively. Then, the top part of our algorithm is as the two formula,

$$B(t,m) = \min_{j=1}^{Nn}\left[D(t-1,j) + C_{cross}\left(\mathrm{SB}(t-1,j), \mathrm{SF}(t,m)\right)\right] \quad (1)$$

$$D(t,n) = \min_{k=1}^{Nm}\left[B(t,k) + C_{inner}\left(\mathrm{SF}(t,k), \mathrm{SB}(t,n)\right)\right] \quad (2)$$

where $B(t,m)$ denotes the minimum accumulated cost for the m-th LHS candidate at time t, and $D(t,n)$ denotes the minimum accumulated cost for the n-th RHS candidate at time t. In addition, C_{cross}(X, Y) is the cost function defined here to calculate the cost introduced by the two cross-syllable units, X (a RHS unit for the predecessor syllable) and Y (a LHS unit for the current syllable), and C_{inner}(U, V) is the cost function defined here to calculate the cost introduced by the two within-syllable units, U (an LHS unit for the current syllable) and V (a RHS unit for the current syllable). The definitions for the two cost functions are given in Subsections 3.2 and 3.3.

3.2. Within-syllable cost function

When an LHS unit and a RHS unit is to be connected to form a syllable, we think three types of costs should be considered, i.e. transformation cost, continuity cost, and concatenation cost. Here, "transformation" includes pitch and duration transformations to let a selected unit's pitch and duration satisfy its corresponding note's requirement. "continuity" means phonetic continuity between an LHS unit and a RHS unit to be connected whereas "concatenation" means spectral concatenation at the boundary between an LHS unit and a RHS unit. Therefore, we define the within-syllable cost function, C_{inner}(X, Y), as the formula,

$$\begin{aligned} &C_{inner}\left(\mathrm{SF}(t,m), \mathrm{SB}(t,n)\right) = \\ &C_{tran-p}\left(\mathrm{SF}(t,m), \mathrm{SB}(t,n)\right) + \alpha \times C_{tran-d}\left(\mathrm{SF}(t,m), \mathrm{SB}(t,n)\right) \\ &+\beta \times C_{conti}\left(\mathrm{SF}(t,m), \mathrm{SB}(t,n)\right) + \gamma \times C_{con}\left(\mathrm{SF}(t,m), \mathrm{SB}(t,n)\right) \end{aligned} \quad (3)$$

where α, β, and γ are the weighting constants. Here, we set α to 0.1, β to 0.8, and γ to 0.1 empirically.

3.3. Cross-syllable cost function

When two adjacent syllables are to be connected, we think two types of costs should be considered, i.e. continuity cost and concatenation cost. Here, "continuity" means contextual continuity between a RHS unit of the predecessor syllable and an LHS unit of the current syllable. "concatenation" still means spectral concatenation at the boundary between a RHS unit and an LHS unit. Therefore, we define the cross-syllable cost function, C_{cross}(X, Y), as the formula,

$$\begin{aligned} C_{cross}\left(\mathrm{SB}(t-1,n),\mathrm{SF}(t,m)\right) = \\ \lambda \times C_{contc}\left(\mathrm{SB}(t-1,n),\mathrm{SF}(t,m)\right) + \\ \eta \times C_{con}\left(\mathrm{SB}(t-1,n),\mathrm{SF}(t,m)\right) \end{aligned} \tag{4}$$

where λ and η are the weighting constants. We set λ to 1.2 and η to 0.01 empirically.

3.4. Pitch transformation cost

The pitch transformation cost, $C_{tran\text{-}p}$(SF(t, m), SB(t, n)), used in Formula (3) is calculated as the formula,

$$\begin{aligned} C_{tran-p}\left(\mathrm{SF}(t,m),\mathrm{SB}(t,n)\right) = PtchCost\left(\mathrm{SF}(t,m),\mathrm{Note}(t)\right) \\ + PtchCost\left(\mathrm{SB}(t,n),\mathrm{Note}(t)\right) \end{aligned} \tag{5}$$

where Note(t) denote the note corresponding to the t-th syllable, and the function, $PtchCost(X, Y)$ is used to calculate the pitch-difference cost between X and Y. Let $PD(X, Y)$ denotes the pitch difference, in semitones, between X and Y. Then, we define the function, $PtchCost(X, Y)$, as the formula,

$$PtchCost\left(X,Y\right) = \begin{cases} 0, & \text{if } PD(X,Y)=0 \\ 2\times PD(X,Y)-1, & \text{if } PD(X,Y)<3 \\ (PD(X,Y))^2, & \text{if } PD(X,Y)>=3 \end{cases} \tag{6}$$

The idea of Formula (6) is to let the pitch-difference cost grow exponentially in order to prevent a unit of large pitch difference from being selected. We think pitch difference is correlated with the difference in formant frequencies.

3.5. Duration transformation cost

The duration transformation cost, $C_{tran\text{-}d}$(SF(t, m), SB(t, n)), used in Formula (3) is calculated as the formula,

$$\begin{aligned} C_{tran-d}\left(\mathrm{SF}(t,m),\mathrm{SB}(t,n)\right) = \left|DurPlan(t,0)-Dur(\mathrm{SF}(t,m))\right| \\ + \left|DurPlan(t,1)-Dur(\mathrm{SB}(t,n))\right| \end{aligned} \tag{7}$$

where $DurPlan(t,0)$ denotes the planned duration, in frames, for the LHS unit of the t-th syllable, $DurPlan(t,1)$ denotes the planned duration for the RHS unit of the t-th syllable, and $Dur(X)$ denotes the actual duration, in frames, of the candidate unit, X.

3.6. Concatenation cost

The concatenation cost, $C_{con}(X, Y)$, used in both Formula (3) and (4) is calculated as the formula,

$$\begin{aligned} C_{con}\left(X,Y\right) = 0.5\times dist\left(Last(X),Frst(Y)\right) \\ + 0.5\times dist\left(Next(X),Scnd(Y)\right) \end{aligned} \tag{8}$$

where $Last(X)$ denotes the last frame of X, $Next(X)$ denotes the next to last frame of X, $Frst(Y)$ and $Scnd(Y)$ denote the first and second frames of Y respectively, and $dist(x, y)$ denotes the geometric distance between the two DCC vectors, x and y.

3.7. Continuity cost

Let the label tuple for the RHS unit, SB(t-1, n), be <Uc, Up, Un>, the label tuple for the LHS unit, SF(t, m), be <Vc, Vp, Vn>, and $ClssNo(x)$ denote the class number of x. Then, the continuity cost, C_{contc}(SB(t-1, n), SF(t, m)), used in Formula (4) is calculated as the formula,

$$\begin{aligned} C_{contc}\left(\mathrm{SB}(t-1,n),\mathrm{SF}(t,m)\right) = \\ \left|ClssNo(Uc)-ClssNo(Vp)\right| + \left|ClssNo(Un)-ClssNo(Vc)\right| \end{aligned} \tag{9}$$

4. Perceptual evaluation

To evaluate the unit selection algorithm proposed, we have conducted two listening tests. The first test is to compare the fluency levels of the synthesized songs by our unit selection algorithm but under two different criteria. The second test is to measure the MOS scores of the synthesized songs' qualities. Here, we invite 26 persons to participate the two listening tests. Among the participants, only 6 of them are familiar with the research field of speech processing.

Singing fluency we think is majorly affected by spectral continuities around the boundaries of adjacent units. For the test of fluency level, one faster-tempo song and one slower-tempo song are synthesized under two different criteria respectively. One of the criteria is as specified in Formula (1) and (2), i.e. cost minimized dynamic programming (DP). By contrast, the other criterion is to

replace the minimization in Formula (1) and (2) with maximization, i.e. cost maximized DP. For each of the two songs, two synthesized song versions under different criteria are played to each of the participants, and each participant gives a score to indicate which version is more fluent. In details, the score, -2 (2), would be given if the former version is definitely more (less) fluent than the latter. The score, -1 (1), would be given if the former version is slightly fluent (influent) than the latter. Otherwise, the score, 0, is given if the fluency level cannot be distinguished. After listening test, the scores given by the 26 participants are arranged and averaged. As a result, we obtain the average (AVG) scores and standard deviations (STD) listed in Table 1. According to the average scores in Table 1, it is apparently that our unit selection algorithm (cost minimized DP) can indeed select better sequence of demi-syllable units to have synthesized songs to be more fluent. We think the average score would become better if most of the participants are familiar with speech processing and do not give conservative scores.

TABLE 1. AVERAGE SCORES FOR FLUENCY LEVEL

Song	AVG	STD
A (slower tempo)	1.154	1.347
B (faster tempo)	0.923	1.093

The second listening test is to evaluate the MOS scores of the synthesized songs' qualities (considering both fluency and naturalness). The two songs used in the first listening test are also used here. For each of the two songs, the song version synthesized under cost maximized DP is taken as the reference for the MOS score of one point. By contrast, the song version recorded from a real singer is taken as the reference for the MOS score of five points. Then, the song version synthesized under cost minimized DP is used as the test song and a participant is requested to give a score to indicate the quality of this synthesized song. After listening test, the scores given by the 26 participants are collected and averaged. As a result, we obtain the average scores and standard deviations listed in Table 2. According to the

TABLE 2. AVERAGE MOS SCORES FOR SONG QUALITY

Song	AVG	STD
A (slower tempo)	3.577	0.945
B (faster tempo)	3.192	1.234

average scores in Table 2, it can be seen that out unit selection algorithm (cost minimized DP) can improve the synthesized songs' quality from 1 point to around 3.38 points. Therefore, our unit selection algorithm is effective in improving the qualities of the synthesized songs. In addition, the improving in song quality is more noticeable for a synthesized song of slower tempo as shown by the average scores, 3.577 vs. 3.192.

4. Conclusions

In this study, we propose a DP based unit selection algorithm for singing voice synthesis. The formula for this algorithm are derived in a top-down structure, and hence we think such formula are noticeably nice. The voice unit adopted here is demi-syllable. Such unit, demi-syllable, is intend to overcome the situation that only a limited quantity of training-songs are available. For splitting a syllable into two demi-syllable units, we have studied a heuristic method which is shown to be workable according to the results of the listening tests.

To evaluate the unit selection algorithm proposed, we have conducted two listening tests for spectral fluency and song quality, respectively. In the test of spectral fluency, our method (cost minimized DP) obtains the average score, 1.04, which means the spectral discontinuities (occurring in some boundaries of adjacent units) can be significantly reduced. In the test of song quality, our method obtains the average MOS score, 3.38, which indicates the qualities of the songs synthesized by our method can be notably improved.

References

[1] H. Y. Gu and H. L. Liao, "Mandarin singing-voice synthesis using an HNM based scheme", Journal of Information Science and Engineering, Vol. 27, No. 1, pp. 303-317, Jan. 2011.

[2] J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models", IEEE Signal Processing Magazine, Vol. 24, No. 2, pp. 67-79, March 2007.

[3] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, "Expression control in singing voice synthesis: features, approaches, evaluation, and challenges", IEEE Signal Processing Magazine, Vol. 32, No. 6, pp. 55-73, Nov. 2015.

[4] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, "Recent development of the HMM-based singing voice synthesis system—Sinsy", in Proc. ISCA Workshop on Speech Synthesis, Kyoto, Japan, pp. 211-216, Sept. 2010.

[5] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0", in Proc. of ISCA Workshop on Speech Synthesis, Bonn, Germany, pp. 294-299, Aug. 2007.

[6] N. Schnell, G. Peeters, S. Lemouton, P. Manoury, and X. Rodet, “Synthesizing a choir in real-time using pitch synchronous overlap add”, Int. Computer Music Conference, Berlin, Germany, pp. 102-108, 2000.

[7] H. Y. Gu and Z. F. Lin, "Singing-voice synthesis using ANN vibrato-parameter models", Journal of Information Science and Engineering, Vol. 30, No. 2, pp. 425-442, March 2014.

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

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

[10] M. Umbert, J. Bonada, and M. Blaauw, "Generating singing voice expression contours based on unit selection", Stockholm Music Acoustics Conference, Stockholm, Sweden, pp. 315-320, July 2013.

[11] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system", INTERSPEECH, Pittsburgh, Pennsylvania, pp. 2274-2277, Sept. 2006.