JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 30, 425-442 (2014)

Singing-voice Synthesis Using ANN Vibrato-parameter Models*

HUNG-YAN GU AND ZHENG-FU LIN
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan

Vibrato is an important factor that affects the naturalness level of a synthetic singing voice. Therefore, the analysis and modeling of vibrato parameters are studied in this paper. The vibrato parameters of those syllables segmented from recorded songs are analyzed by using short-time Fourier transform and the method of analytic signal. After the vibrato parameter values for all training syllables are extracted and normalized, they are used to train an artificial neural network (ANN) for each type of vibrato parameter. Then, these ANN models are used to generate the values of vibrato parameters. Next, these parameter values and other music information are used together to control a harmonic-plus-noise model (HNM) to synthesize Mandarin singing voice signals. With the synthetic singing voice, subjective perception tests are conducted. The results show that the singing voice synthesized with the ANN generated vibrato parameters is much increased in the naturalness level. Therefore, the combination of the ANN vibrato models and the HNM signal model is not only feasible for singing voice synthesis but also convenient to provide multiple singing voice timbres.

Keywords: singing voice, vibrato parameter, pitch contour, analytic signal, artificial neural network

1. INTRODUCTION

The techniques of singing voice synthesis may be used to construct a tutoring system for singing, a virtual singer, or a part for an entertainment system. Currently, several techniques have been proposed to synthesize singing voice signals, including phase vocoder [1, 2], formant synthesis [1, 2], LPC synthesis [1, 3], sinusoidal model [4], PSOLA synthesis [5], EpR (excitation plus resonances) model [6, 7], and corpus-based synthesis [8, 9]. Also, we had proposed an HNM (harmonic-plus-noise model) based and improved scheme to synthesize a Mandarin singing-voice signal [10]. Nowadays, to synthesize a clear (not noisy and not reverberant) singing voice signal is not difficult. Nevertheless, the synthesized singing voice is usually not felt as natural and expressive as that sung by a real singer even though some performance rules [11] are already adopted. We think one of the major reasons is that the factors relevant to the expressing of singing voice are not adequately modeled and controlled. Such factors include vibrato, marcato, soffocato, rubato, *etc.* Among these factors, vibrato is thought to be the most important one. Therefore, in this paper, we studied to analyze and model the parameters of vibrato. Hope that the synthesized singing voice can present natural expression of vibrato.

According to the studies by Horii [12] and Imaizumi, *et al.* [13], the most notable

Received January 14, 2012; revised March 19 & May 1, 2012; accepted June 5, 2012.
Communicated by Hsin-Min Wang.
* The preliminary version has been presented in 2008 International Conference on Machine Learning and Cybernetics, July 12-15, 2008, Kunming, China and it was supported by National Science Council, Taiwan, under Grant No. NSC 96-2218-E-011-002.

phenomenon due to vibrato is that the pitch-frequency will vibrate quasi-periodically. An example is as the solid-lined curve in Fig. 1, which is obtained from analyzing a real sung syllable. In this figure, the pitch contour is seen to vibrate between 295Hz and 315Hz, and the vibrating rate is about 4.9Hz. Therefore, to synthesize singing voice with vibrato expression, the pitch contour is the major acoustic factor to deal with.

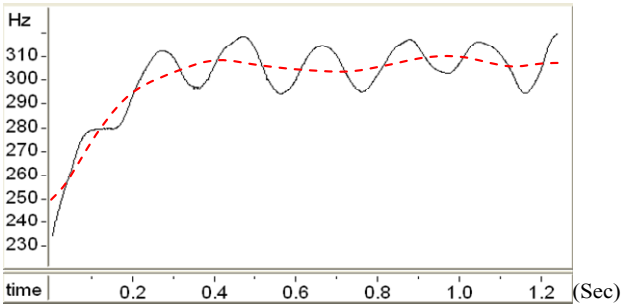

Fig. 1. Pitch contour analyzed from a real sung syllable.

Although a vibrating pitch contour may be generated by applying some rules [1, 11], its naturalness level is usually not as natural as that expressed by a real singer. Note that vibrato is not only presented in the fast vibration part within the solid-lined curve in Fig. 1 but also presented in the slow vibration as the dash-lined average-pitch curve in Fig. 1. That is, the average-pitch curve is also very influential to naturalness-level perception especially at the left and right ends, which reflect the contextual effects. Additionally, according to the studies by Sundberg, *et al.* [14], and Shonle and Horan [15], a vibrating pitch contour can be analyzed and represented with three types of parameters, *i.e.* intonation, vibrato extent, and vibrato rate. Intonation means the smoothed (or averaged) pitch contour as the dash-lined curve in Fig. 1, which is called the slow vibration here. Vibrato extent is the deviation of the vibration (*e.g.* peak value minus intonation value), and vibrato rate is the peak-valley variation rate on the fast vibrating pitch contour.

Therefore, we decide to model the vibrato parameters with ANN. Here, the ANN based models are not used to generate a vibrating pitch contour directly but used to generate its corresponding vibrato parameters. In terms of the generated vibrato parameters, a pitch contour that expresses vibrato can then be indirectly generated. Afterward, the generated pitch contour is used to determine the pitch-tuned HNM parameters' values for each control point placed on the time axis of the singing syllable to be synthesized [10]. Then, the singing voice signal of vibrato expression can be synthesized by using the HNM based signal synthesis scheme studied previously [10]. In addition, through the use of HNM, multiple singing-voice timbres can be conveniently provided for a user to select. This is because the addition of a new timbre only requires that the 408 syllables of Mandarin Chinese are recorded once from a new speaker and then analyzed to obtain their HNM parameters. HNM is originally proposed by Y. Stylianou [16, 17]. It may be viewed as improving the sinusoidal model [18] to better model the noise signal components in the higher frequency band of a voice signal.

In the following, the methods adopted to analyze the vibrato parameters will be explained in Section 2. The details about modeling vibrato parameters with ANN will be

described in Section 3. Then, in Section 4, the methods adopted to synthesize pitch contour and singing voice signal will be explained. Also, the perception tests conducted are described. Finally, concluding remarks are given in Section 5.

2. VIBRATO PARAMETER ANALYSIS

In this paper, vibrato parameters were analyzed from a real singer's singing voice and then used to train the ANN models. In detail, we follow the steps of the flowchart in Fig. 2 to do vibrato parameter analysis and ANN model training. First, song signals sung by a real singer are recorded. Secondly, the recorded signals are labeled manually with phonetic symbols and segmented to a separate signal file for each sung syllable. For each syllable's signal, its instantaneous pitch frequency (IPF) curve is measured next. The meaning of IPF is the instantaneous frequency of the first harmonic partial as mentioned in others' works [19, 20]. Then, the IPF curve is further analyzed to extract the intonation, vibrato extent, and vibrato rate parameters. The processing steps mentioned above will be detailed in the following subsections. As to the training of the ANN models, explanations will be given in Section 3.

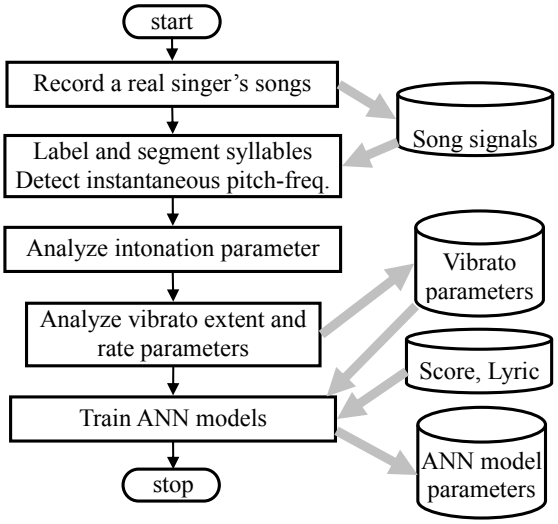

Fig. 2. Processing steps of the training stage.

2.1 Recording Singing-voice Signals

We invited a male student singer to sing several popular Mandarin songs in a soundproof room. He followed the MIDI accompaniment played to his headphone. Hence, the pitch of each lyric syllable sung should be in tune with the accompaniment. Singing signals were recorded in real-time, *i.e.* signal samples were directly saved to a computer file, and the sampling rate is 22,050Hz. As a total, 15 songs sung in different days were recorded, and the total number of segmented lyric syllables is 2,841. Among the 15 songs,

the tempos range from 72 to 120 beats per minute, *i.e.* slower and quicker songs are both included.

2.2 Measuring Instantaneous Pitch Frequency

For Mandarin songs, a lyric syllable usually has only one music note assigned to it. Hence, syllable is taken as the voice unit. Here, the lyric syllables of a song are labeled manually with the software, WaveSurfer [21], and then segmented into separate signal files. Note that a Mandarin syllable may be started with an unvoiced initial consonant but the syllable final part is always voiced. Therefore, the boundary point between the unvoiced and voiced segments must be determined first with a pitch detection method. Then, the curve of IPF is measured after the boundary point. Here, we calculate both autocorrelation function and absolute magnitude difference function to do pitch detection.

The method adopted to measure IPF is as the following. First, the voiced segment is sliced into a sequence of frames. The length of each frame is 512 sample points but the frame shift is only 32 sample points. For each frame, the signal samples are Hamming windowed, and appended with zero valued samples in order to perform 4,096 points FFT (fast Fourier transform). Then, on the FFT spectrum, the leading five harmonic peaks are searched from 0Hz with the method proposed by Stylianou [16]. Let $g(i)$ denote the frequency value (in Hz) of the ith harmonic peak. For each $g(i)$ found, it is divided by i to give an estimate of fundamental frequency. Then, the five estimates are geometrically averaged to give an IPF value for this frame. When the IPF values of all frames are obtained, they are connected to form an IPF curve. This IPF curve, $f(t)$, is fitted here with the time-varying function [19],

$$f(t) = V_d(t) + V_e(t) \cdot \cos(\phi(t)) \tag{1}$$

where $V_d(t)$ represents its intonation parameter, $V_e(t)$ represents its vibrato-extent parameter, and its vibrato-rate parameter, $V_r(t)$, can be derived as

$$V_r(t) = \frac{1}{2\pi} \cdot \frac{d\phi(t)}{dt}. \tag{2}$$

2.3 Analysis of Intonation Parameter

A simple idea to obtain the intonation parameter curve, $V_d(t)$, is to low-pass filter the IPF curve, $f(t)$. Low-pass filtering may be done in the frequency domain or time domain. Here, we select to filter the IPF curve in the time-domain with a moving-average filter. This is because we intend to keep the global curve shape, and which can be achieved by introducing fixed time delays for all frequencies through moving-average filtering. In more details, at a time point t, the IPF values, $f(\tau)$, $\tau = t - 128, t - 127, \ldots, t + 128$, are averaged to get the intonation parameter value, $V_d(t)$.

2.4 Analysis of Vibrato Extent and Rate

To obtain the curves of vibrato extent $V_e(t)$ and vibrato rate $V_r(t)$, the signal $s(t)$ that

is defined here as

$$s(t) = V_e(t) \cdot \cos(\phi(t)) = f(t) - V_d(t) \tag{3}$$

is computed first according to Eq. (1). Then, by using the analysis method of analytic signal [22], $V_e(t)$ and $\phi(t)$ can be derived consequently.

Suppose that the analytic signal of $s(t)$ is $z(t)$. Then, according to Gabor's definition [22], $z(t)$ is a complex signal and is composed of the real part, $s(t)$, and the imaginary part, $\hat{s}(t)$. That is,

$$\begin{aligned} z(t) &= s(t) + j \cdot \hat{s}(t), \\ \hat{s}(t) &= H[s(t)] = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau, \end{aligned} \tag{4}$$

where $H[s(t)]$ denotes Hilbert transform [19, 22, 23]. Hilbert transform can rotate the phase angle of the signal with the right amount of $\pi/2$. Consequently, we obtain that

$$\hat{s}(t) = V_e(t) \cdot \sin(\phi(t)) \tag{5}$$

$$z(t) = V_e(t) \cdot \exp(j \cdot \phi(t)). \tag{6}$$

Then, $V_e(t)$ and $\phi(t)$ can be derived as

$$V_e(t) = \sqrt{s^2(t) + \hat{s}^2(t)}, \tag{7}$$

$$\phi(t) = \text{atan}(\hat{s}(t)/s(t)). \tag{8}$$

In terms of $\phi(t)$, vibrato rate parameter, $V_r(t)$, can be computed according to Eq. (2).

For practical implementation, Hilbert transform in Eq. (4) can be done with a more efficient method [20, 23]. Suppose the signal sequence, $s(t)$, has N signal samples. The first step of the method is to apply DFT (discrete Fourier transform) to $s(t)$ to obtain its long-term spectrum, $S(k)$, $k = 0, 1, ..., N-1$. Next, for the frequency bins in the first half, their amplitudes are doubled, *i.e.* let $Z(k) = 2 \cdot S(k)$ for $k = 0, 1, \ldots, N/2 - 1$. For the frequency bins in the second half, their amplitudes are however directly set to zero, *i.e.* let $Z(k) = 0$ for $k=N/2, N/2 + 1, ..., N - 1$, in order to make the signal analytic. Then, in the third step, apply inverse DFT to $Z(k)$, $k = 0, 1, ..., N - 1$, to obtain its time-domain complex signal sequence, $z(t) = s(t) + j \cdot \hat{s}(t)$. Then, the imaginary part of the complex signal sequence, $z(t)$, would be the desired Hilbert-transformed signal sequence, $\hat{s}(t)$.

3. ANN VIBRATO PARAMETER MODELS

In last section, the methods adopted to analyze the three vibrato parameters are presented. In practice, besides intonation $V_d(t)$, vibrato extent $V_e(t)$, and vibrato rate $V_r(t)$, we need one more parameter, *i.e.* the initial phase $\phi(0)$, in order to have the initial pitch frequency be correctly generated. Therefore, we decide to train an ANN for each of the four vibrato parameters. ANN models are adopted here in the hope that the singing style, in

expressing vibrato, of the invited singer can be learned. Here, each ANN is actually a multi-layer perceptron (MLP) [24]. The adopted learning algorithm is back propagation. The structure of each MLP is as the one shown in Fig. 3. That is, only one hidden layer is placed between the input and output layers. Within each node located at the hidden or output layers, the hyperbolic tangent function,

$$f(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \tag{9}$$

is adopted as the transformation function because the values of the vibrato parameters may be negative or positive. The number of nodes in the output layer is 32 for three of the MLPs but the MLP for initial phase needs only one output node. The details for vibrato parameter representation and normalization are given in Section 3.1. As for the input layer, the contextual data of the current lyric syllable to be sung are fed. The details of the contextual data used here are given in Section 3.2. For the number of nodes to be placed in the hidden layer, some experiments have been done. The details of the experiment results are given in Section 3.3.

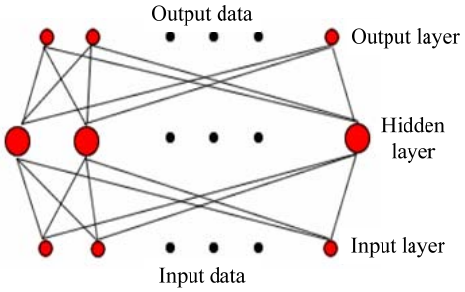

Fig. 3. The structure of the MLP.

3.1 Vibrato Parameter Sampling and Normalization

Note that the time lengths of the intonation curves (or vibrato extent and rate curves) analyzed from different lyric syllables may be very different. Nevertheless, these curves must all be used to train the intonation MLP. Therefore, we have to represent all obtained curves with a predefined number of dimensions according to the number of nodes placed at the output layer. Here, we make a tradeoff between accuracy and computation burden, and select to use 32 nodes for the output layer. Following this number, 32, we adopt a simple representation method that samples a curve at 32 uniformly placed time points. In details, a vibrato parameter's curve, $V_x(t)$, is sampled to $U_x(i) = V_x(T * i/31)$, $i = 0, 1, \ldots, 31$, where T is the time length and the subscript x is used to denote any one of the three types of parameters (intonation, vibrato extent and rate). When a specific vibrato parameter is focused, the subscript x will be changed to d (to denote intonation), e (to denote vibrato extent), or r (to denote vibrato rate).

On the other hand, consider the synthesis of a curve when given 32 output values, $U_x(i)$, from an MLP and a target time length T. A basic idea is to synthesize the curve by means of interpolation. Currently, a simple method of piece-wise linear interpolation is

adopted, which seems enough in practice. In details, for a sample time point t, the intervals $[T_i, T_{i+1}]$, $i = 0, 1, \ldots, 30$ and $T_i = T * i/31$, are searched first to locate the interval $[T_k, T_{k+1}]$ that contains t. Then, the value of the interpolated sample, $V_x(t)$, at time t is computed as

$$V_x(t) = U_x(k) + (U_x(k+1) - U_x(k)) \frac{t - T_k}{T_{k+1} - T_k}. \tag{10}$$

In training a MLP, the 32 sampled values, $U_x(i)$, from a vibrato parameter curve are not used directly as the target values for the MLP to learn. This is because the transformation function defined in Eq. (9) can only output a value ranged from −1 to 1. To suit this value range, the sampled values must be normalized beforehand. Let $U_d(i)$, $i = 0, 1, \ldots, 31$, be sampled from an intonation curve, $V_d(t)$. We first define the normalization factor M_d by taking the geometric mean of those sampled values from the center portion. In details, M_d is defined as

$$M_d = \left(\prod_{i=11}^{20} U_d(i) \right)^{1/10}. \tag{11}$$

The leading and following portions are not used because their sampled values may be unstable due to contextual influences. After the value of M_d is obtained, the sampled values are normalized as

$$\hat{U}_d(i) = \frac{U_d(i)}{M_d} - 1, \; i = 0, \; 1, \; ..., \; 31. \tag{12}$$

Then, the normalized intonation values can only move between −1 and 1 no matter what their original pitch frequencies are.

Let $U_e(i)$, $i = 0, 1, \ldots, 31$, be sampled from a vibrato extent curve, $V_e(t)$. The normalization method adopted here is to divide $U_e(i)$ by $U_d(i)$, *i.e.* $\hat{U}_e(i) = U_e(i)/U_d(i)$, to let the normalized extent values become relative to intonation. As to the curve of vibrato rate, its sampled values, $U_r(i)$, are normalized here by dividing them with the constant 20, *i.e.* $\hat{U}_r(i) = U_r(i)/20$, since the value of $U_r(i)$ may not be greater than 20. As to the parameter of initial phase, $\phi(0)$, its value is normalized by dividing with the constant 5.

3.2 Contextual Information and Their Classification

What factors is the expressing of vibrato affected by? We think the factors include (a) the note duration, syllable-initial type, and syllable-final type of the current syllable to be sung, (b) the note duration and syllable-final type of the previous syllable, (c) the note duration and syllable-initial type of the next syllable, and (d) the pitch-height differences between the current note and its previous and next notes. Since the number of factors considered here is not small, the number of possible combinations of these factors' values will be very huge. However, the training data used here include just 15 songs that have only 2,841 syllables in total. Therefore, classification of these factors' values is

inevitably needed in order to reduce the number of possible combinations.

Among the three duration factors, the current note's duration is thought to be more important than the two adjacent notes' durations. Therefore, we decide to divide the current note's duration into 5 classes but to divide the adjacent notes' durations into just 3 classes. For the current note, the 5 classes are defined as 0 to 0.3 sec., 0.3 to 0.5 sec., 0.5 to 0.8 sec., 0.8 to 1.3 sec., and above 1.3 sec. For the adjacent notes, the 3 classes are defined as 0 to 0.25 sec., 0.25 to 0.5 sec., and above 0.5 sec. Thus, 3 bits and 2 bits are needed to represent their class indices, respectively.

For the two syllable-final factors, the 39 syllable-final types of Mandarin Chinese are divided into 4 classes. These classes are single vowel (*e.g.* /a/), diphthong (*e.g.* /ai/), triphthong (*e.g.* /iau/), and nasal-ended final (*e.g.* /ang/). Also, for the two syllable-initial factors, the 21 syllable-initial types of Mandarin Chinese are divided into 3 classes. These classes are voiced consonants (*e.g.* the nasals and liquids), short unvoiced consonants (*e.g.* the non-aspirated stops), and long unvoiced consonants (*e.g.* the aspirated stops and fricatives). Therefore, syllable initial and final classes need 2 bits, respectively, to represent their indices.

As to the two factors of pitch-height differences, 7 classes are defined here. The pitch-height difference is calculated in semitones. The elements of the 7 classes are as listed in Table 1. To distinguish these classes, 3 bits are used to represent the class indices.

Table 1. Classes of pitch-height differences.

Class	1	2	3	4	5	6	7
Elements (semitone)	–6, –7, –8,…	–3, –4, –5	–1, –2	0	1, 2	3, 4, 5	6, 7, 8,…

About the details of the contextual data to be fed to an MLP, let us consider the leading four lyric syllables of the song, "O Susanna", as an example. The lyric syllables are /wo/, /lai/, /zii/, and /a/. The assignment of notes according to the score is that the first two notes, <do, 0.1875 sec.> and <re, 0.1875 sec.>, are assigned to /wo/, <mi, 0.375 sec.> is assigned to /lai/, <sol, 0.375 sec.> is assigned to /zii/, and <sol, 0.5625 sec.> is assigned to /a/. Since /wo/ has two notes assigned to, we will prepare two sets of contextual data for successive feeding to the MLPs to generate two pitch contours for the two notes. As to the next two syllables, we will prepare one set of contextual data for each. The details of the input data sets prepared are as those listed in Table 2. In Table 2, the abbreviations, "Pre.", "dur.", "Cur.", "Post.", "syll.", and "diff.", represents "Previous", "duration", "Current", "Posterior", "syllable", and "difference", respectively. Previous, current, and posterior note durations are the three duration factors as mentioned earlier. Previous syllable's final and current syllable's final are the two factors of syllable finals whereas current syllable's initial and posterior syllable's initial are the two factors of syllable initials. In addition, previous pitch difference means the pitch-height difference (in semitones) calculated as the current note's pitch minus the previous note's pitch. Similarly, posterior pitch difference means the pitch-height difference calculated as the posterior note's pitch minus the current note's pitch.

Table 2. Example contextual data sets to be fed to the MLPs.

Data set	Lyric	Note	Pre. note dur.	Cur. note dur.	Post. note dur.	Pre. syll. final	Cur. syll. final	Cur. syll. initial	Post. syll. initial	Pre. pitch diff.	Post. pitch diff.
1	/wo/	do	00	000	00	00	01	10	10	100	101
2	/wo/	re	00	000	01	01	01	10	10	101	101
3	/lai/	mi	00	001	01	01	01	10	00	101	110
4	/zii/	sol	01	001	01	01	00	00	10	110	100

3.3 Experiments for MLP Training

In training each MLP, the initial value of the learning rate is set to 2. Then, each time a training iteration is completed, the learning rate is multiplied with the factor, 0.95, to decrease its influence. Also, according to empirical experience, the number of training iterations is set to 1,500, which is large enough to let the node-connection weight vector converge well.

Table 3. Prediction errors of the intonation MLP.

Number of nodes	AVG	STD	MAX
6	0.03786	0.03374	0.33116
8	0.03754	0.03366	0.33151
10	0.03936	0.03395	0.32476
12	0.03875	0.03375	0.32874
16	0.03941	0.03437	0.32827

Table 4. Prediction errors of the vibrato extent MLP.

Number of nodes	AVG	STD	MAX
6	0.01237	0.01523	0.18382
8	0.01215	0.01492	0.17955
10	0.01237	0.01522	0.18291
12	0.01255	0.01545	0.17809
16	0.01257	0.01547	0.17915

For each MLP, the number of nodes to be placed in the hidden layer needs to be determined according to the results of the training experiments. Therefore, we have tried to place 6, 8, 10, 12, and 16 nodes to the hidden layer, respectively, in different runs of the training program. Here, the prediction error of a lyric syllable's vibrato parameter is calculated in RMS (root mean square) manner. According to all training syllables' prediction errors, three error statistics are calculated next, *i.e.* average of prediction error (AVG error), standard deviation of prediction error (STD error), and maximum of prediction error (MAX error). For the intonation MLP, its error statistics obtained when placing different number of nodes to the hidden layer are as listed in Table 3. In addition, for the vibrato-extent MLP, its error statistics obtained are as listed in Table 4.

According to the error statistics obtained in training the four vibrato parameter MLPs, it can be said that the prediction errors do not change considerably when different

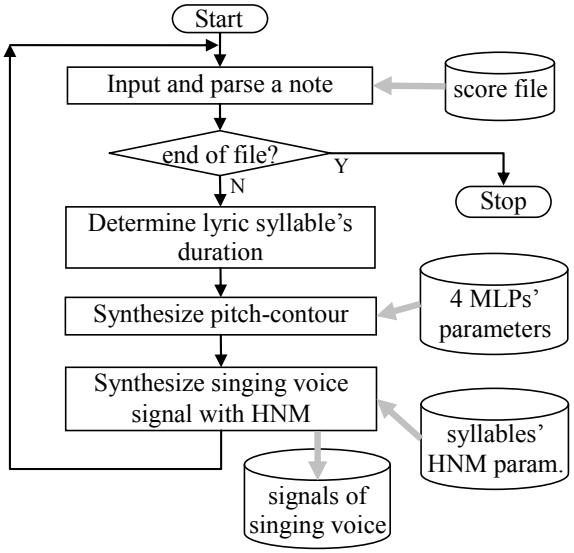

Fig. 4. Main flow of the singing voice synthesis system.

number of nodes are placed to the hidden layer. Therefore, we decide to place 8 nodes to the hidden layer for each of the four MLPs.

4. SINGING VOICE SYNTHESIS AND PERCEPTION TEST

Integrating the four vibrato parameter MLPs for generating pitch contours, we have constructed a Mandarin singing voice synthesis system that is able to express vibrato. The main processing flow of this system is shown in Fig. 4. In the first block, a music note's information is parsed from a text line of a score file. According to the parsed item of beats and the global parameter of tempo, the duration for the parsed lyric syllable to be sung can be computed. Next, the contextual data are gathered and fed to the four vibrato-parameter MLPs. According to the vibrato parameter values predicted by the MLPs, a vibrato-expressing pitch contour can then be generated. The details for generating a pitch contour are explained in Sections 4.1 and 4.2. Afterward, in the last block of Fig. 4, the pitch contour is used to adjust the lyric syllable's HNM parameters. Then, the adjusted HNM parameters are used to synthesize a singing voice signal with an HNM based and improved method [10]. This method will be explained in Section 4.3.

4.1 Pitch Contour Generation

When the contextual data of a lyric syllable are fed to the four MLPs, sampled and normalized vibrato parameter values will be predicted and available on the output-layers of the MLPs. As the next step, inverse normalizations are performed according to the formula inversed from the normalization formula near Eq. (12). Then, the sampled vibrato parameters, $U_d(i)$, $U_e(i)$, $U_r(i)$, and initial phase $\phi(0)$, are restored to their correct scale.

To synthesize an intonation curve, the pitch frequency (in Hz), F, of the current note is needed, and F can be looked up in terms of the current note's pitch symbol (*e.g.* "G3"). Also, the duration, T, of the lyric syllable is needed, which is already computed in the second block of Fig. 4. By replacing M_d in Eq. (12) with F, the sampled intonation parameters, $U_d(i)$, can be computed as $U_d(i) = (\hat{U}_d(i) + 1) \cdot F$. Such $U_d(i)$ obtained would have the correct pitch. Next, by interpolating $U_d(i)$ with Eq. (10) and the duration T, an intonation curve, $V_d(t)$, can be obtained.

When the sampled intonation parameters, $U_d(i)$, are ready, the sampled vibrato extent parameters, $U_e(i)$, can be computed as $U_e(i) = \hat{U}_e(i) \cdot U_d(i)$. Then, the vibrato extent curve, $V_e(t)$, can be interpolated with Eq. (10) and the duration T. Similarly, the vibrato rate curve, $V_r(t)$, can be obtained by interpolating the sampled parameters, $U_r(i)$, with Eq. (10) and T.

After the curves, $V_d(t)$, $V_e(t)$, and $V_r(t)$, are generated, the phase curve, $\phi(t)$, is next generated in terms of $V_r(t)$ as

$$\phi(t) = \phi(t-1) + 2\pi \cdot V_r(t) \cdot \frac{1}{22{,}050}, \quad t = 1, 2, \ldots, T-1. \tag{13}$$

where 22,050 is the sampling rate. Finally, the pitch contour, $P(t)$, can be generated as

$$P(t) = V_d(t) + V_e(t) \cdot \cos(\phi(t)), \quad t = 0, 1, \ldots, T-1. \tag{14}$$

A lyric syllable may sometimes be assigned two notes, which means it should be sung in portamento. In this paper, each note has a pitch contour generated for it. Therefore, we have to merge the two pitch contours generated for a syllable sung in portamento. A merging method studied here is as the following. First, the two pitch contours are each divided into three segments of equal time lengths. Secondly, the first segment of the first pitch contour is taken as the leading segment for the final pitch contour whereas the third segment of the second pitch contour is taken as the tailing segment. Next, the middle segment of the final pitch contour is generated by using the two boundary values of this segment and a cosine-based interpolation method [10].

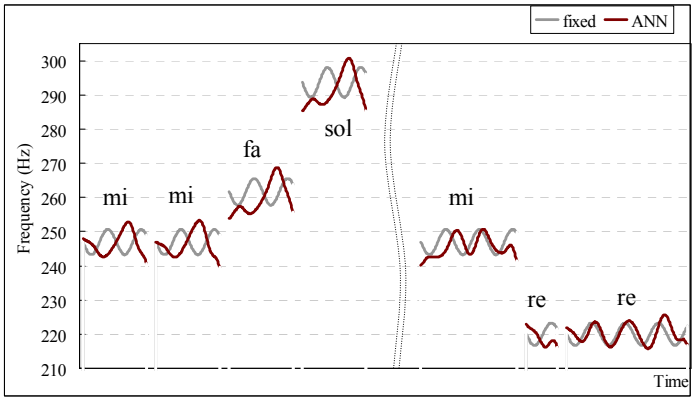

Fig. 5. Example synthetic pitch contours for the song ode-to-joy.

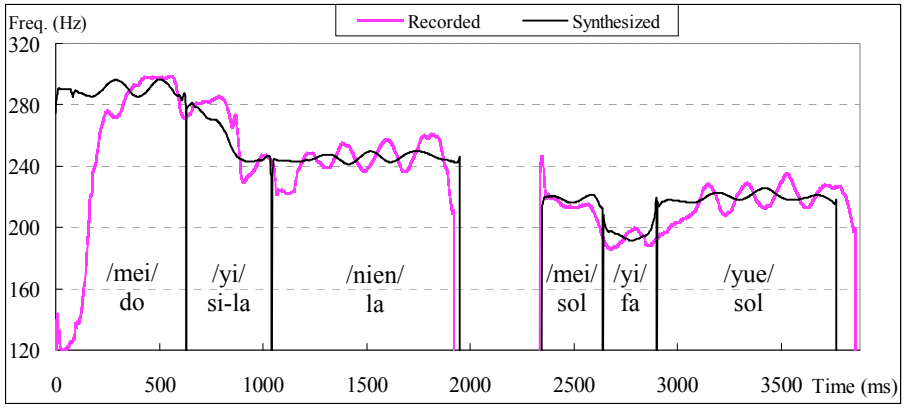

Fig. 6. Comparison between synthesized and recorded pitch contours.

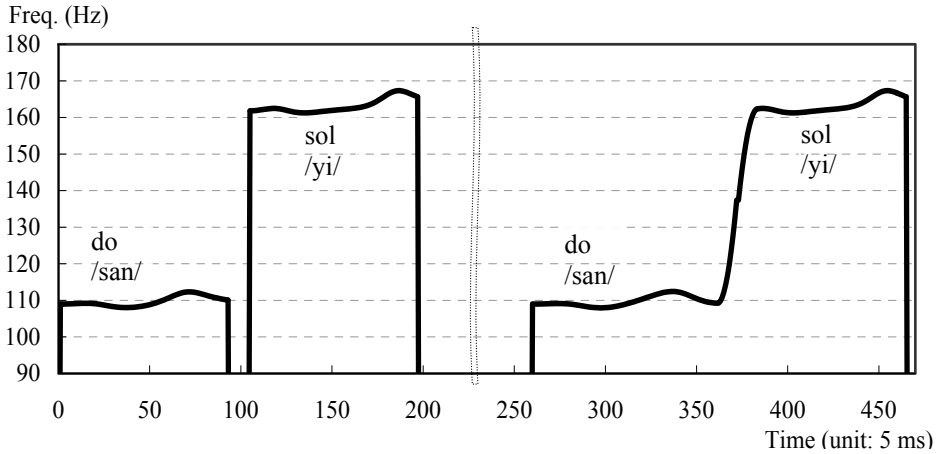

Fig. 7. An example of pitch co-articulation.

To show the generated pitch-contours, here we take the first sentence of the song, Ode to Joy (not recorded to train the MLPs), as an example. The melody of the sentence is <mi, mi, fa, sol, …, mi, re, re>. After applying the generation procedure given above, we obtain the seven heavily drawn pitch contours shown in Fig. 5 whereas the seven lightly drawn pitch contours are obtained by setting some constant values (as those for synthesizing SB in Section 4.4) to the vibrato parameters. From Fig. 5, it can be found that every pair of pitch contours are of very different curve shapes except the pair for the last note. Also, it can be seen that the pitch contours generated with the MLPs vibrate strongly in their extents. Nevertheless, the perceived pitches of these notes are all in tune. In addition, the melody consisting of these notes is felt of much higher naturalness level.

For another example of synthesized pitch contours, we feed the contextual data of of a recorded song (*i.e.* used in training the models) to the MLPs here. The first seven notes are <do, si, la, la, sol, fa, sol>. They are assigned to the six lyric syllables, /mei/, /yi/, /nien/, /mei/, /yi/, and /yue/, where the second and third notes, <si> and <la>, are both

assigned to the second syllable /yi/. As a result, we obtain the six pitch contours as those heavily drawn in Fig. 6. For comparison, we also have the pitch contours of the corresponding notes in the recorded song analyzed and then lightly drawn in Fig. 6. From Fig. 6, it can be seen that the vibrato extents of the recorded pitch contours are noticeably large (more than 20Hz) for the syllables, /nien/ and /yue/. Nevertheless, the vibrato extents of the synthesized pitch contours are relatively small even though the extent of /nien/ is already larger than 8.1 Hz. The reasons for why the MLPs do not generate vibrato extents as large as those sung by the real singer we think include the following two points. First, the value of every contextual data type is grouped here to a few classes due to insufficient training songs. Secondly, the singer who is invited to record songs may use larger or smaller vibrato extents as his emotional expressions when singing different songs of different music genres. Therefore, the MLPs can only learn the averaged vibrato characteristics from the songs recorded from the singer.

4.2 Pitch Co-articulation

Pitch co-articulation is meant that the pitch contours of two adjacent syllables are smoothly connected across the syllable boundary. An example of pitch co-articulation is shown in Fig. 7. The two pitch contours at the left side of Fig. 7 are disconnected whereas the two at the right side are smoothly connected, *i.e.* pitch co-articulated. It is seen that the pitch contours of some lyric syllables are connected to their predecessor syllables in the songs sung by a real singer. Therefore, to synthesize more natural singing voice, we cannot always place a short pause between every two synthetic singing syllables, or directly connect the pitch contours of adjacent syllables to be pitch co-articulated. Otherwise, a synthetic song will be perceived as a sequence of isolated lyric syllables according to our listening experiences. To demonstrate this point, we have prepared a web page, http://guhy.csie.ntust.edu.tw/vibrato/PitchCoa.html, from which synthetic songs with and without pitch co-articulation can be downloaded and compared.

According to the knowledge of articulatory phonetics, pitch co-articulations will occur if the duration of the predecessor note is short (*e.g.* less than 0.7 sec.), and the syllable final of the predecessor syllable and the syllable initial of the successor syllable are both consisted of voiced phonemes. For example, there may be a pitch co-articulation between the two syllables, /san/ and /ming/, if /san/ is sung in a short duration. Nevertheless, pitch co-articulation will not occur between the two syllables, /yang/ and /sia/, since /s/ is unvoiced. To synthesize pitch co-articulation, the first step is to have the defined note duration for the predecessor syllable fully taken, *i.e.* do not leave a short pause and generate a pitch contour for the entire duration. Next, eliminate the last segment, 50 ms in length, of the predecessor syllable's pitch contour, and also eliminate the first segment, 50 ms in length, of the successor syllable's pitch contour. Here, 50 ms is selected according to listening to the songs synthesized under different settings of time lengths. Then, the two remaindered pitch contours can be directly connected with a line segment. It may be worried that a line segment will cause slope discontinuities at the two ends of the line segment. Nevertheless, the effect caused by slope discontinuities is hardly perceivable according to listening to the synthetic songs. In fact, we obtain a good perceptual effect that a synthetic song will be much improved in its continuity and naturalness level.

4.3 Singing-voice Signal Synthesis

Note that Mandarin Chinese is a syllable prominent language and there are only 408 different syllables when the lexical tones are not distinguished. Therefore, we recorded and saved each of the 408 syllables just once for analyzing its HNM parameters. The HNM parameters for a signal frame include the frequency, amplitude, and phase of each harmonic partial in the lower frequency band, and 20 linear cepstrum coefficients used to approximate the higher frequency band's spectral envelope. Here, two speakers (one female and one male) were invited to pronounce the 408 syllables in isolation in a sound proof room, respectively. Note that none of the two speakers is the singer who sung the 15 songs for training the MLPs. This design, separating the training of the vibrato model and the analyzing of the HNM parameters, gives an advantage that a new singing-voice timbre can be added to our system with just a small effort. The effort is to record the 408 syllables of Mandarin Chinese from a new speaker. In contrary, a large effort will be required for a corpus-based singing-voice synthesis system to add a new timbre.

Since each syllable has only one utterance, it is not possible to do unit selection here. Therefore, the signals for a lyric syllable under various combinations of pitch heights and durations must all be synthesized in terms of the same analyzed HNM parameters for that syllable. Then, two problems are inevitably encountered. That is, the timbres of the synthetic syllable signals must be kept consistent when the original (or recorded) pitch contour is tuned to some requested target pitch contours that are generated with Eq. (14) and pitch co-articulation. Secondly, the synthetic syllable signals must be as fluent as possible when the original syllable duration is lengthened or shortened. These two problems were already studied and a feasible solution method is presented in our previous work [10]. Our method is different from that proposed by Stylianou [16, 17]. For the purpose of keeping timbre consistent, we proposed and used a Lagrange-interpolation based local approximation method to estimate the spectral envelope on the lower frequency band. This method is efficient and seems enough according to our experiences of listening to some synthesized songs. In addition, for the problem of lengthening or shortening a syllable's duration, we propose and use a kind of piece-wise linear time-mapping function. Such mapping function can reduce the duration of the starting or ending voiced consonant of a syllable in order to synthesize a more fluent sung syllable signal.

As to the detailed operations for signal synthesis, the signal sample located at time t is synthesized as the harmonic signal, $H(t)$, plus the noise signal, $N(t)$. $H(t)$ is synthesized as

$$H(t) = \sum_{k=0}^{L} a_k^n(t) \cos(\phi_k^n(t)), \quad t = 0,\ 1,\ ...,\ 100, \tag{15}$$

where L is the number of harmonic partials, 100 is the number of samples between the nth and $(n+1)$th control points, $a_k^n(t)$ is the time-varying amplitude of the kth partial at time t, and $\phi_k^n(t)$ is the cumulated phase for the kth partial at time t. In our system, $a_k^n(t)$ and $\phi_k^n(t)$ are just linearly varied as,

$$a_k^n(t) = A_k^n + \frac{t}{100}(A_k^{n+1} - A_k^n), \tag{16}$$

$$\phi_k^n(t) = \phi_k^n(t-1) + 2\pi \cdot f_k^n(t) \big/ 22{,}050\,, \tag{17}$$

$$f_k^n(t) = F_k^n + \frac{t}{100} \cdot (F_k^{n+1} - F_k^n)\,, \tag{18}$$

where A_k^n and F_k^n represent the amplitude and frequency of the kth harmonic partial on the nth control point.

As to $N(t)$, it is also synthesized as a summation of sinusoidal components similar to Eq. (15). Nevertheless, these sinusoids occupy the higher frequency band, adjacent sinusoids are always placed 100 Hz apart, and their frequencies do not change with time. In addition, the amplitudes of these sinusoids are still linearly varied with time. Therefore, on a control point, the amplitudes of the sinusoids need to be determined according to the 20 cepstrum coefficients.

4.4 Perception Tests

Two song scores, "Ode to Joy" and "Kang-Ding madrigal", are used, respectively, for two runs of perception tests. Each song score is used to synthesize three singing voice files. The first file denoted with SA is synthesized with no vibrato. This can be accomplished by setting $U_d(i) = F$ and $U_e(i) = 0$. The second file denoted with SB is synthesized with fixed vibrato parameter values. That is, we set $U_d(i) = F$, $U_e(i) = F * 1.5 / 100$, and $U_r(i) = 4$. As to the third file, it is denoted with SC and its vibrato parameters are generated by the four MLPs constructed here. Then, the three files are played as the two pairs, (SA, SB) and (SB, SC), to each of the 15 invited participants to perform perception tests. Each participant is requested to give two scores of naturalness comparison, *i.e.* comparing SB with SA and comparing SC with SB. A score of 0 is defined if the naturalness level between two files cannot be distinguished. A score of 1 (or −1) is defined if the latter (or former) played is slightly better. In addition, a score of 2 (or −2) is defined if the latter (or former) played is sufficiently better.

After the scores given by the participants in the two runs of tests are collected, the average score and standard deviation are computed to be 0.73 and 0.93 for comparing SB with SA. For comparing SC with SB, the average score and standard deviation are computed to be 0.57 and 1.09. According to the average score, 0.73, it is seen that the naturalness level can be increased even fixed vibrato parameter values are adopted. Also, according to the average score, 0.57, using MLPs to generate vibrato parameter values can indeed help synthesizing more natural singing voice. As our opinion, since most of the participants (actually 10 persons) taking part in the perception tests are not familiar with the research field of singing voice synthesis, the average scores should be increased a lot if the participants are all familiar with this research field. For demonstration, the web page, http://guhy.csie.ntust.edu.tw/vibrato/, is prepared which can be accessed to download the three synthetic singing voice files, SA, SB, and SC.

5. CONCLUDING REMARKS

Vibrato is commonly found in real singing voices as a way for expressing music mood. Therefore, the modeling of vibrato styles and the generation of vibrato parameter

values are important issues for a computer to synthesize natural and expressive singing voice. In this paper, we study to analyze, represent, and normalize the four types of vibrato parameters, *i.e.*, intonation, initial phase, and vibrato extent and rate. For a vibrato-parameter curve, 32 uniformly sampled data are adopted to represent it. Then, the sampled data are normalized by using the formula studied here. In addition, we propose to use an MLP to model each type of vibrato parameter, *i.e.* training the MLP with the analyzed, sampled, and normalized vibrato-parameter data.

According to the practical measurement experiments, short-time Fourier transform based instantaneous pitch frequency estimation and the analysis method of analytic signal are found to be feasible for analyzing vibrato parameter. After the MLPs for the four types of vibrato parameters were trained, we have integrated them into our previous HNM based Mandarin singing voice synthesis system. With the integrated system, singing voice files are synthesized under different conditions to conduct perception tests. According to the result of the perception tests, the singing voice synthesized by using the MLP generated vibrato parameters can indeed be much increased in the naturalness level. This may verify that the slow vibration, *i.e.* the intonation curve, is also very influential to the perceived naturalness level. In addition, the combination of the MLP vibrato models and the HNM signal model is not only feasible for singing voice synthesis but also convenient to provide multiple singing voice timbres for a user to select.

REFERENCES

1. F. R. Moore, *Elements of Computer Music*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
2. C. Dodge and T. A. Jerse, *Computer Music: Synthesis*, *Composition*, *and Performance*, Schirmer Books, NY, 1997.
3. G. A. Frantz, K. S. Lin, and K. M. Goudie. "The application of a synthesis-by-rule system to singing," *IEEE Transactions on Consumer Electronics*, Vol. 28, 1982, pp. 257-262.
4. M. W. Macon, L. Jensen-Link, J. Oliverio, M. A. Clements, and E. B. George, "A singing voice synthesis system based on sinusoidal modeling," in *Proceedings of International Conference on Acoustics*, *Speech*, *and Signal Processing*, 1997, pp. 435-438.
5. N. Schnell, G. Peeters, S. Lemouton, P. Manoury, and X. Rodet, "Synthesizing a choir in real-time using pitch synchronous overlap add," in *Proceedings of International Conference on Computer Music Conference*, 2000, pp. 102-108.
6. J. Bonada and A. Loscos, "Sample-based singing voice synthesizer by spectral concatenation," in *Proceedings of Stockholm Music Acoustics Conference*, 2003, pp. 1-4.
7. J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE Signal Processing Magazine*, Vol. 24, 2007, pp. 67-79.
8. Y. Meron, "High quality singing synthesis using the selection-based synthesis scheme," Ph.D. Dissertation, Department of Information and Communication Engineering, University of Tokyo, 1999.
9. C.-Y. Lin, T.-Y. Lin, and J.-S. R. Jang, "A corpus-based singing voice synthesis system for Mandarin Chinese," in *Proceedings of the 13th ACM International Conference on Multimedia*, 2005, pp. 359-362.

10. H. Y. Gu and H. L. Liao, "Mandarin singing-voice synthesis using an HNM based scheme," *Journal of Information Science and Engineering*, Vol. 27, 2011, pp. 303-317.
11. X. Rodet, "Synthesis and processing of the singing voice," in *Proceedings of the 1st IEEE Benelux Workshop on Model Based Processing and Coding of Audio*, 2002, pp. 99-108.
12. Y. Horii, "Acoustic analysis of vocal vibrato: a theoretical interpretation of data," *Journal of Voice*, Vol. 3, 1989, pp. 36-43.
13. S. Imaizumi, H. Saida, Y. Shimura, and H. Hirose, "Harmonic analysis of the singing voice: acoustic characteristics of vibrato," in *Proceedings of Stockholm Music Acoustics Conference*, 1994, pp. 197-200.
14. J. Sundberg, E. Prame, and J. Iwarsson, "Replicability and accuracy of pitch patterns in professional singers," *Vocal Fold Physiology*, *Controlling Complexity and Chaos*, P. J. Davis and N. H. Fletcher, ed., Singular Publishing Group, San Diego, 1996.
15. J. I. Shonle and K. E. Horan, "The pitch of vibrato tones," *Journal of Acoustical Society of America*, Vol. 67, 1980, pp. 246-252.
16. 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.
17. Y. Stylianou, "Modeling speech based on harmonic plus noise models," in *Nonlinear Speech Modeling and Applications*, G. Chollet *et al.*, eds., Springer-Verlag, Berlin, 2005, pp. 244-260.
18. T. F. Quatieri, *Discrete-Time Speech Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
19. I. Arroabarren, M. Zivanovic, J. Bretos, A. Ezcurra, and A. Carlosena, "Measurement of vibrato in lyric singers," *IEEE Transactions on Instrumentation and Measurement*, Vol. 51, 2002, pp. 660-665.
20. H. Suzuki, F. Ma, H. Izumi, O. Yamazaki, S. Okawa, and K. Kido, "Instantaneous frequencies of signals obtained by the analytic signal method," *Acoustical Science and Technology*, Vol. 27, 2006, pp. 163-170.
21. WaveSurfer, http://www.speech.kth.se/wavesurfer/index.html, Centre for Speech Technology, Kungliga Tekniska högskolan, Stockholm, Sweden.
22. H. G. Feichtinger and T. Strohmer, *Gabor Analysis and Algorithms: Theory and Applications*, Birkhauser, Boston, 1998.
23. A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
24. K. Gurney, *An Introduction to Neural Networks*, UCL Press, London, 1997.

Hung-Yan Gu (古鴻炎) received the B.S. and M.S. degrees in Computer Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively, and the Ph.D. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1990. Currently, he is a Professor in the Department of Computer Science and Information Engineering, National Taiwan University of Science

and Technology, Taipei, Taiwan. His research interests include speech signal processing, computer music synthesis, and information hiding.

Zheng-Fu Lin (林正甫) was born in 1981. He received the B.S. degree in Computer Science and Engineering from Yuan Ze University, Taoyuan, Taiwan, in 2005, and the M.S. degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2008.