An HMM Based Pitch-contour Generation Method for Mandarin Speech Synthesis

HUNG-YAN GU AND CHUNG-CHIEH YANG*
Department of Computer Science and Information Engineering
**Institute of Electrical Engineering*
National Taiwan University of Science and Technology
Taipei, 106 Taiwan

In this paper, a method is proposed to generate pitch-contours for Mandarin speech synthesis. In this method, an HMM (hidden Markov model) is used to model the prosodic states implicitly stayed and a syllable's pitch-contour is treated as an observation generated from a prosodic state. Such an HMM is called a syllable pitch-contour HMM (SPC-HMM). For training the SPC-HMM, we developed a feasible method to normalize a pitch-contour's height. After normalization, each training syllable's pitch-contour is vector quantized and represented with a VQ (vector quantization) code. Then, the VQ code and its adjacent syllables' lexical tones are combined to define an observation symbol for training the SPC-HMM. In the synthesis phase, a sentence-wide most probable observation symbol sequence is searched on the SPC-HMM using a dynamic programming algorithm proposed here. Then, the observation symbol found for a syllable is decoded to obtain its pitch-contour VQ code. We conducted testing experiments to determine the size of a pitch-contour codebook and the number of states for an SPC-HMM. The results indicate that setting the codebook size to eight and using six states are the best choices. Also, we conducted perception tests to compare the naturalness levels of synthetic speech files. The results show that the two generation modes for operating an SPC-HMM studied here are comparable to each other in naturalness level.

Keywords: speech synthesis, pitch contour, pitch normalization, hidden Markov model, vector quantization

1. INTRODUCTION

A Mandarin TTS (text-to-speech) system is conventionally decomposed into three main processing components, *i.e.*, text analysis, prosodic parameter generation, and signal waveform synthesis [1]. When a Chinese sentence is input, it will first be analyzed by the text analysis component to determine its corresponding sequence of syllables and lexical tones. Note that Mandarin is a tonal language, and a tone shape (*e.g.*, falling, rising, or leveling) superimposed on a syllable carries lexical information. After textual analysis, the prosodic parameters, pitch-contour, duration, amplitude, and pre-pause for each syllable are decided by the prosodic-parameter generation component. According to the generated prosodic parameters, the signal synthesis component is then invoked to synthesize speech signals. Recently, the component of prosodic parameter generation was paralleled with a newly added component, spectrum-progression parameter generation [2]. Furthermore, other researchers not only have modeled spectrum progression with an HMM but also have integrated the function of prosodic parameter generation into an extended HMM [3]. Nevertheless, it is not known whether modeling both spectrum pro-

gression and prosodic parameter generation simultaneously with an extended HMM is the best choice.

In previous studies, we have proposed two signal synthesis methods, TIPW [4] and HNMES [5]. TIPW (time proportioned interpolation of pitch waveform) is a time-domain method that can reduce the drawbacks, chorus and reverberation, found in PSOLA [6]. In contrast, HNMES (HNM extended scheme) is a frequency-domain HNM (harmonic-plus-noise model) [7] based and extended method that can synthesize speech signals with higher clarity than TIPW or PSOLA. Nevertheless, the naturalness level of synthetic speech is dominantly determined by the generated pitch-contours. Therefore, many researchers have spent effort to study the generation of pitch-contours.

Intonation models for pitch-contour generation were previously classified into two categories, *i.e.* tone sequence models and superposition models [8]. Among the two, the concept of superposition has been much more influential for later studies on generating Mandarin syllables' pitch-contours with a representative superposition model being the Fujisaki model [9]. Recently, several methods have been proposed to generate Mandarin syllables' pitch-contours. One of them is the use of concatenation rules [10]. Another two methods are based on regression analysis [11] and regression trees [12]. Also, artificial neural network based methods have been proposed [13, 14]. Finally, another notable statistical method is the one proposed by Lai [15].

Besides the methods mentioned, HMM (hidden Markov model) based methods have also been proposed to generate pitch-contours [3, 16, 17]. The method proposed by Ljolje and Fallside [16], however, only considered the generation of a pitch-contour for an isolated syllable, and the states of the HMM built for a particular tone are transited frame by frame. Although the method proposed by Fukada, *et al.* [17] can generate a pitch-contour for each comprising syllable of a sentence, their HMMs are actually built for context dependent phones and the states of an HMM are still transited frame by frame. In the more recent study by Tokuda, *et al.* [3], a context dependent phone's pitch contour is generated using just its corresponding HMM states' pitch statistics, and its adjacent phones' pitch information is not considered and used. Therefore, in those methods, pitch-contours are generated with just local (within a phone or syllable) optimization consideration without the sentence-wide global concept of prosodic states. Consequently, we were motivated to consider a different HMM based pitch-contour generation method that will take sentence-wide optimization into account and will adopt syllable instead of signal-frame as the time unit for state transition [18].

As mentioned in [19], a syllable at the start of a sentence is usually uttered with higher pitch than one at the end, *i.e.*, the phenomenon of declining. Considering this phenomenon, we imagine that there are three prosodic states that occupy sentence-initial, sentence-middle, and sentence-final parts respectively. Nevertheless, we do not know how to assign a sentence's syllables to these states explicitly. Therefore, these prosodic states are treated as the hidden states of an HMM, and the transitions between the prosodic states are restricted to have a left-to-right structure [19, 20]. Besides the influence of a syllable's stayed prosodic state, the lexical tone of the syllable and the lexical tones of its adjacent syllables have considerable influence on the syllable's pitch-contour shape and height. Thus, the lexical-tones of a syllable and its adjacent syllables must also be modeled. To model these factors within an HMM, we propose encoding these factors' combinations as observation symbols of a discrete HMM. Since these factors' values are discrete, a discrete HMM is a more direct selection than a continuous HMM. The detail

of observation symbol encoding will be described in Section 2.4. As to other minor factors, such as syllable-final type and syllable position within a word, they are not modeled in this study. This is because only a limited number, 375, of training sentences are recorded currently. Here, the HMM based pitch-contour model is called a syllable pitch-contour HMM (SPC-HMM).

For training the SPC-HMM, the main processing flowchart shown in Fig. 1 is followed. The pitch-contours obtained from analyzing the training sentences must first be normalized on both time and pitch-height. In this paper, we develop a feasible method for pitch-height normalization. After normalization, a pitch-contour codebook for each lexical tone is trained, and the pitch-contour of each syllable is then vector quantized with the codebook trained for the syllable's lexical tone. Then, the VQ code of a syllable's pitch-contour and the lexical tones of the syllable and its adjacent syllables are combined to define an observation-symbol. Afterward, the observation-symbol sequence obtained from each training sentence is used to train the SPC-HMM.

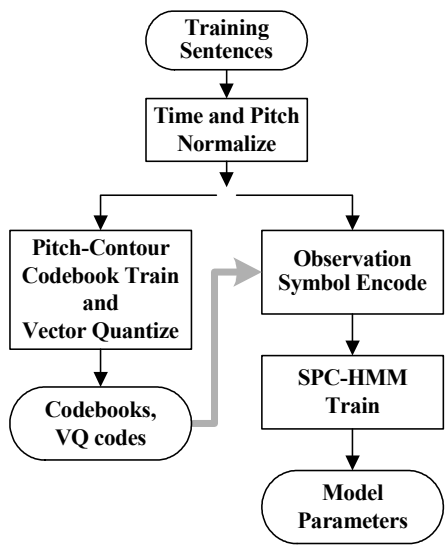

Fig. 1. Main flow for training SPC-HMM.

As for generating pitch-contours, the main processing flowchart shown in Fig. 2 is followed. An input sentence first will be text analyzed to obtain the necessary information, *i.e.* the syllable and lexical tone of each Chinese character. According to the lexical tones, candidate observation symbols can be encoded. Then, a sentence-wide most probable observation-symbol sequence is searched in the SPC-HMM using the dynamic-programming based algorithm proposed here. Next, the observation symbol found for a syllable is decoded to get its pitch-contour VQ code. In terms of the VQ code, the pitch-contour can then be looked up from the corresponding VQ codebook.

In Section 2, the functions of the blocks in Fig. 1 are explained in detail. Then, in Section 3, the functions of the blocks in Fig. 2 are explained. In Section 4, experiments for determining the size of a VQ codebook and the number of states for an SPC-HMM are described, and the pitch-contours generated by different SPC-HMMs are evaluated

with subjective perception tests. Finally, conclusions are given in Section 5.

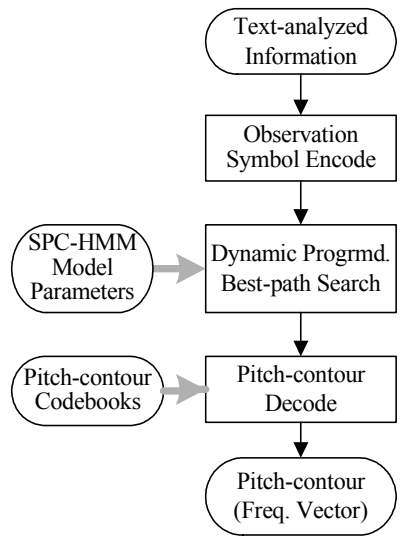

Fig. 2. Main flow for generating pitch-contours.

2. SPC-HMM TRAINING

2.1 Pitch-contour Representation

Legendre polynomials [21] and Chebyshev polynomials [22] have been proposed to expand a syllable's pitch-contour for speech coding and pronunciation assessment applications, respectively. In contrast, direct representation by sampling a pitch-contour [12] and expanded representations with the coefficients of Legendre polynomials [13, 15] or cosine functions [23] have been adopted by different researchers to construct their own pitch-contour generation models. Also, another type of representation that uses just one line segment to represent a syllable's pitch contour is notable [24].

According to the study by Ravuri and Ellis [24], listeners would prefer the original pitch contours to the linear approximated contours in only 60% of cases. Therefore, sampling a syllable's pitch-contour with 16 points would be more than adequate. Nevertheless, considering that Mandarin is a tonal language, we still choose to represent a pitch-contour with a vector of 16 pitch frequencies. These pitch frequencies are computed at 16 normalized time points that are placed uniformly over a syllable's voiced segment. For a normalized time point, its pitch frequency is obtained by interpolating the four pitch frequencies measured in the four successive signal frames surrounding this time point. In more detail, consider the example shown in Fig. 3. The point labeled t on the horizontal-axis is the concerned normalized time point, and the unit of time is signal sample. Suppose that the two closest signal frames on the left side are located at time points t_k and t_{k+1} and have the measured pitch frequencies, f_k and f_{k+1}, whereas the two closest signal frames on the right side are located at time points t_{k+2} and t_{k+3} and have the measured pitch frequencies, f_{k+2} and f_{k+3}. Then, the pitch frequency, f_t, for the time point, t, is com-

puted with the Lagrange interpolation formula [25]:

$$f_t = \sum_{i=0}^{3} f_{k+i} \cdot \left(\prod_{\substack{j=0 \\ j \neq i}}^{3} \frac{t - t_{k+j}}{t_{k+i} - t_{k+j}} \right) . \tag{1}$$

To measure the pitch frequency of a signal frame, an autocorrelation based method [19] was adopted. In order to verify the measured pitch frequencies, manual checking and correcting were performed after automatic measuring.

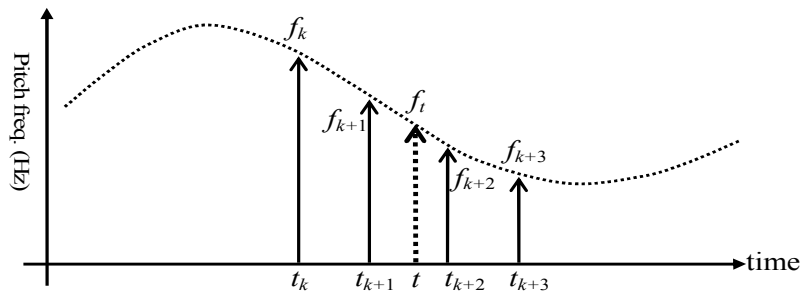

Fig. 3. An example of pitch-frequency interpolation.

2.2 Pitch-height Normalization

Training sentences are usually recorded across several days under different moods. Hence, the pitch height of a training sentence may deviate considerably from the gross mean. If pitch height normalization is not performed, abnormal pitch-contour transitions between some adjacent syllables will be heard in the synthesized speech. Therefore, pitch height normalization must be done before training the SPC-HMM. For this reason, we developed a simple but feasible normalization method. Using this method, each training sentence just needs to be recorded once. That is, it is not necessary to record a training sentence several times and pick the one of the desired pitch height. When developing the pitch height normalization method, we noted that any two adjacently recorded sentences did not have contextual influence to each other since they were randomly selected from different articles and there was a long break between their recordings. Therefore, we need not consider the contextual influence of adjacent sentences on a given sentence's average pitch height. The processing steps of the proposed normalization method are:

(a) For the i-th training sentence, compute its j-th syllable's average pitch-height E_{ij} in logarithmic scale. That is,

$$E_{ij} = \frac{1}{16} \sum_{k=0}^{15} g_{ijk} , \quad g_{ijk} = \log(f_{ijk}) , \tag{2}$$

where f_{ijk} represents the pitch frequency obtained on the k-th normalized time point of the j-th syllable. Then, compute this sentence's average pitch-height U_i as

$$U_i = \frac{1}{n_i} \sum_{j=1}^{n_i} E_{ij} \ , \tag{3}$$

where n_i denotes the number of syllables in the i-th training sentence.

(b) Compute the gross average pitch-height, UA**,** across all training sentences,

$$UA = \frac{1}{SM} \sum_{i=1}^{SM} U_i \quad , \tag{4}$$

where SM denotes the number of recorded training sentences.

(c) Compute the pitch-height modification value, Z_i, for the i-th training sentence as

$$Z_i = U_i - UA \tag{5}$$

(d) According to Z_i, normalize the pitch contour of the j-th syllable in the i-th training sentence as

$$\overline{g}_{ijk} = g_{ijk} - Z_i \quad , \quad k = 0, 1, ..., 15, \quad j = 1, 2, ..., n_i \quad , \tag{6}$$

Although the pitch height normalization method given above seems simple, it can however eliminate most abnormal pitch-contour transitions found between adjacently generated syllable pitch-contours. To further smooth the transition between two adjacently generated syllable pitch-contours, we studied another pitch-height normalization method. This method is applied to the resultant pitch-contours obtained from the prior normalization method. The processing steps of this method are:

(e) Uniformly divide each training sentence into three segments. Then, collect the syllables that are divided into the first segment from all training sentences. Next, compute the average pitch height, α_k^1, from the collected syllables that are pronounced in the k-th lexical tone. Similarly, the average pitch heights, α_k^2 and α_k^3, can be computed from those syllables that are divided into the second and third segments.

(f) For the i-th training sentence, compute its j-th syllable's pitch-height deviation, d_{ij}. Then, compute the average deviation, $\overline{d}_i$, for this sentence. That is,

$$d_{ij} = E_{ij} - \alpha_k^m \ , \quad m = \left\lfloor \frac{(j-1)}{n_i} \cdot 3 \right\rfloor + 1, \quad j = 1, 2, \ ..., n_i \ , \tag{7}$$

$$\overline{d}_i = (d_{i,1} + d_{i,2} + ... + d_{i,n_i}) / n_i \tag{8}$$

where E_{ij} is the pitch height of the j-th syllable computed from the prior normalization method, m is the segment number that the j-th syllable is divided into, k is the lexical-tone number of the j-th syllable, and n_i is the number of syllables in the i-th training sentence.

(g) According to the average deviation, $\bar{d}_i$, for the *i*-th training sentence, normalize the pitch-contour of the *j*-th syllable in this sentence as

$$\bar{g}_{ijk} = g_{ijk} - \bar{d}_i \quad , \quad k = 0, 1, ..., 15, \quad j = 1, 2, ..., n_i \ , \tag{9}$$

where g_{ijk}, *k*=0,1, …, 15, is the *j*-th syllable's pitch-contour obtained from the prior normalization method.

2.3 Vector Quantization of Syllable Pitch-contour

After pitch height normalization, the pitch-contour of the *j*-th syllable in the *i*-th training sentence would be represented as a vector of 16 frequency values, $\bar{g}_{i,j,0}$, $\bar{g}_{i,j,1}$, …, $\bar{g}_{i,j,15}$, in logarithmic scale. To model a syllable pitch-contour's frequency vector as an observation of a discrete HMM, we must vector quantize it beforehand. We know that the technique, vector quantizing a pitch-contour, has been proposed previously by other researchers. In the tone recognition study [26], pitch and delta-pitch frequencies of each frame are vector quantized while in the speech coding study [21], a syllable's pitch-contour is vector quantized as a whole.

Another consideration here for vector quantizing a syllable pitch-contour is the following. A Mandarin syllable may be superimposed with one of five lexical tones, *i.e.* high-leveling, mid-rising, mid-falling, high-falling, and neutral. When a lexical tone is superimposed to a syllable, a particular pitch-contour is instantiated for the lexical tone. Nevertheless, the pitch height and contour shape of a pitch-contour is not only affected by the current lexical tone but also affected by the immediately preceding and following syllables' lexical tones, *i.e.* the contextual effect. Accordingly, the pitch-contours instantiated for a lexical tone may be of very different pitch heights and contour shapes. Therefore, we decide to cover these pitch-height and contour-shape variations of a particular lexical tone's pitch-contours with vector quantization.

Here, the pitch-height normalized pitch-contours are first divided into 5 sets according to the lexical tones that they represent. Then, for each lexical tone's pitch-contours, we use GLA (Generalized Lloyd Algorithm) to train a VQ codebook [27]. The distance measure adopted here is an RMS (root mean square) one, *i.e.*,

$$dist(y, w) = \sqrt{\frac{1}{16} \sum_{k=0}^{15} (y_k - w_k)^2} \tag{10}$$

Apparently, the quantization error will become smaller when a larger codebook is adopted. This, however, is not always good because a larger codebook size will result in larger observation space for the SPC-HMM. This larger observation space means the estimated HMM parameter will be coarser. Therefore, a tradeoff should be made.

In this paper, 375 training sentences were recorded from a male speaker for training pitch-contour VQ codebooks and training the SPC-HMM. The text of each training sentence was independently and randomly selected from different articles. The guideline for

selection is that every possible tone combination of three adjacent syllables should be found in both initial and final parts of some sentences. The textual details of these sentences can be seen at http://guhy.csie.ntust.edu.tw/PitchCntr/ 375.html. Among the syllables of the training sentences, 632, 754, 462, 867, and 210 of them are uttered in high-leveling, mid-rising, mid-falling, high-falling, and neural tones, respectively. Using the recorded training sentences, we tested three conditions about pitch-contour codebook training, *i.e.*, no pitch height normalization, normalization using the first method, and normalization using the two methods given above. When the codebook size is 8, the VQ errors measured are on average, 0.03337, 0.03012, and 0.02925, respectively, for the three conditions. That is, VQ error will become smaller as more normalization methods are executed.

2.4 Observation Symbol Encoding

In the SPC-HMM, the three hidden states are intended to model the hidden prosodic states. Next, consider how to define the observation symbols for the SPC-HMM. Note that the height and shape of a concerned syllable's pitch-contour is affected not only by that syllable's lexical tone but also affected by its left and right adjacent syllables' tones. Therefore, at time t (*i.e.* the t-th syllable of a sentence), we decide to combine the t-th syllable's lexical tone and pitch-contour VQ code with its left and right adjacent syllables' lexical tones to define the observation symbol, O_t, for the time point t. That is,

$$\begin{aligned} O_t &= encodeA(X_{t-1}, X_t, X_{t+1}, V_t), \quad 0 \le X_t \le 4, \quad 0 \le V_t \le 7 \\ &= 5 \times 5 \times \theta \times X_{t-1} + 5 \times \theta \times X_t + \theta \times X_{t+1} + V_t \end{aligned} \tag{11}$$

where X_t denotes the lexical-tone number of the t-th syllable, V_t denotes the pitch-contour VQ code of the t-th syllable in a training sentence, θ denotes the size of the pitch-contour codebook for X_t, and 5 is the number of different lexical tones. When t=1, X_{t-1} is undefined and must be removed from Eq. (11). Therefore, the encoding function for O_1 is changed to $5\times\theta\times X_t + \theta \times X_{t+1} + V_t + LA$ where $LA = 5\times5\times5\times\theta$. Similarly, when t reaches the last syllable of a sentence, the encoding function for O_t is changed to $5\times\theta\times X_{t-1} + \theta \times X_t + V_t + LA + LB$ where $LB = 5\times5\times\theta$.

In addition, consider that some three-lexical-tone combinations encountered in the synthesis phase may not be seen in the training phase due to the insufficiency of training sentences. We solve this problem by constructing two more simplified SPC-HMMs, for which observation symbols are encoded with fewer factors. That is, the observation symbol encoding functions,

$$\begin{aligned} O_t &= encodeC(X_{t-1}, X_t, V_t) \\ &= LA + 2 \times LB + 5 \times \theta \times X_{t-1} + \theta \times X_t + V_t , \end{aligned} \tag{12}$$

$$\begin{aligned} O_t &= encodeB(X_t, X_{t+1}, V_t) \\ &= LA + 3 \times LB + LC + 5 \times \theta \times X_t + \theta \times X_{t+1} + V_t , \end{aligned} \tag{13}$$

are adopted for the first and second level downgraded SPC-HMMs, respectively, where $LC = 5\times\theta$. Similarly, when t = 1, the encoding function of Eq. (12) is changed to $LA + 3\times LB + \theta \times X_t + V_t$, and when t reaches the last syllable of a sentence, the encoding func-

tion of Eq. (13) is changed to $LA + 4\times LB + LC + \theta \times X_t + V_t$. Then, if an observation symbol encoded with Eq. (11) is not seen in the training sentences, its occurrence probability can still be estimated by alternatively encoding the observation symbol with Eq. (12) or (13) and taking it into the downgraded SPC-HMMs. The second level downgraded model will be tried only if the first level downgraded model cannot be applied.

2.5 SPC-HMM Training

Before the normal and the two downgraded SPC-HMMs can be used to generate syllable pitch-contours, their parameters, a_{ij} (the probability of transiting from state i to state j) and $b_j(k)$ (the probability of observing symbol k at state j), must be trained first. These models can be trained independently since model downgrading will not occur in the training phase. In training these models, we used the algorithm of segmental K-means [20]. The details of the algorithm are found in a relevant textbook. Here, the number of states for an SPC-HMM is tested from 3 to 7, and the state transitions are restricted to a left-to-right manner. The number of recorded training sentences is 375, and the number of syllables comprising these sentences is 2,925. As these quantities of training sentences and syllables are not sufficient, we adopted a sharing (or smoothing) method [28]. That is, when an observation symbol is seen, 0.0009 and 0.0001 of its occurrence probability are shared with the two nearest observation symbols that are encoded with the same lexical tone combination but different pitch-contour VQ code.

In original HMM, observation symbols generated from a same state are assumed to be mutually independent. Nevertheless, within a Mandarin sentence, adjacent syllables' pitch heights are of certain dependency. Therefore, we tried to use a new type of parameter, $c_j(k)$, to model the difference of pitch-height between the prior and current syllables whose lexical tones, X_{t-1} and X_t, are combined to form the observation symbol k at state j. Here, for the t-th syllable of a sentence, its pitch-height difference, $HD(t)$, is defined as

$$HD(t) = HF(t) - HB(t\text{-}1) \tag{14}$$

where $HF(t)$ represents the front pitch-height of the t-th syllable and $HB(t\text{-}1)$ represents the back pitch-height of the $(t\text{-}1)$-th syllable. That is,

$$HF(t) = \frac{1}{8}\sum_{j=0}^{7} g_{t,j} \quad , \quad HB(t) = \frac{1}{8}\sum_{j=8}^{15} g_{t,j} \tag{15}$$

where $g_{t,j}$ is the measured logarithmic pitch frequency on the j-th normalized time point of the t-th syllable. In terms of the parameters $HD(t)$, $c_j(k)$ can be estimated as

$$c_j(k) = \left(\sum_{\substack{t=1 \\ s.t.\ s_t=j \text{ and } O_t=k}}^{n} HD(t) \right) \Bigg/ \left(\sum_{\substack{t=1 \\ s.t.\ s_t=j \text{ and } O_t=k}}^{n} 1 \right) , \tag{16}$$

where the training sentence is assumed to have n syllables, and s_t represents the state stayed by the t-th syllable. According to our experiment results, the average RMS predic-

tion error of a syllable pitch-contour can be improved about 2% if the parameters, $c_j(k)$, are taken into account under setting the codebook size to 8 and setting the number of states to 3. Nevertheless, no improvements were obtained when the codebook size was set to 8 and the number of states was set to 6. Therefore, the new parameter, $c_j(k)$, seems to be insignificant.

3. PITCH CONTOUR GENERATION

When using an HMM for a speech recognition application, a single observation symbol is explicitly defined on each time point (frame). Thus, the search space is just a two dimensional time-state space from which the best (or most probable) path will be found. As to the searching algorithm, a dynamic programming (DP) based algorithm (Viterbi algorithm) is commonly adopted [19, 20]. This, however, is not the case for syllable pitch-contour generation using SPC-HMM. Note that a Mandarin sentence to be synthesized will first be analyzed by the text-analysis component. That is, the syllable sequence corresponding to a Mandarin sentence is already known before pitch-contour generation. Hence, we can encode every three adjacent syllables' lexical tones partially (because the VQ code, V_t, is left to be determined) according to Eq. (11). Since each lexical tone has θ codewords in its pitch-contour VQ codebook, each syllable of the sentence to be synthesized has θ possible encoded observation symbols corresponding to it. Here, the t-th syllable's θ possible observation symbols are denoted as O_t^0, O_t^1, …, O_t^θ. Therefore, when applying SPC-HMM to generate syllable pitch-contours, in addition to the time (syllable) and state axes, a third axis must be added to enumerate the θ possible observation symbol candidates. The addition of the third axis means the conventional DP algorithm for speech recognition cannot be directly applied here.

3.1 Extended Dynamic Programming Algorithm

In this paper, we extend the conventional two-dimensional DP algorithm to solve the three-dimensional DP problem. In an original two-dimensional DP algorithm [20], the term $\delta_t(j)$ is used to denote the probability of the most probable path that will stay at state j on time t. Here, we extend it to $\delta_t(j,k)$ so the third index k can enumerate the θ possible observation symbols. Then, the recursive formula for $\delta_t(j)$ must also be extended. The extension made is:

$$\delta_t(j,k) = \left[\max_{j-1 \le i \le j} \max_{0 \le m \le \theta-1} \delta_{t-1}(i,m) \times a_{ij} \times r(t,j,k,m) \right] \times b_j(O_t^k) \ , \tag{17}$$

$$r(t,j,k,m) \ = \ 1 \Big/ \ exp\left(\left| HF(O_t^k) - HB(O_{t-1}^m) - c_j(k) \right| \right) \ , \tag{18}$$

where i is the index to the prior states, m is the index to the prior observation symbols on time t-1, and $r(t,j,k,m)$ is the probability term introduced here to account for the pitch-height difference between the pitch-contours of the two observation symbols,

O_{t-1}^{m} and O_t^k. Note that the probability of observing symbol k at state j, *i.e.* $b_j(O_t^k)$, may be zero in normal SPC-HMM because of insufficient training data. When $b_j(O_t^k)$ is found to be zero, another observation symbol can be encoded with Eq. (12) or (13) instead and its corresponding occurrence probability is looked up from the corresponding downgraded SPC-HMM. Here, the probability value found from a downgraded SPC-HMM is decreased to one thousandth of its original value before being taken into Eq. (17). This is to prevent model biasing from occurring, *i.e.* preferring a downgraded SPC-HMM. Now, according to Eq. (17), the probability of the most probable path within the three dimensional search space from the starting points, t=1 (time), j=0 (state), and k=0,1,…,7 (pitch-contour VQ code) to the end points, t=n (suppose of n syllables), j=NS (the chosen number of states), and k=0, 1, …, 7, can be computed as:

$$p^* = \max_{0\le k\le 7} \left[\delta_n(NS,k)\right] \tag{19}$$

3.2 Pitch-contour Code Decoding

Actually, we need the sequence of the observation symbols selected along the most probable path but not the probability of the path. If the observation symbol on time t is determined, its corresponding pitch-contour VQ code, V_t, can be decoded according to Eqs. (11), (12), or (13). Next, its corresponding pitch-contour can be found from the VQ codebook of the t-th syllable's lexical tone in terms of V_t. Therefore, backtracking information must be saved during searching the most probable path with Eq. (17). This can be done by saving the values of the index variables, i and m, that maximize the value of $\delta_t(j,k)$ in Eq. (17). Here, we use the variable, $\psi_t(j,k)$, to save the i value of the best coming state, and the variable, $\rho_t(j,k)$, to save the m value of the best prior observation symbol on time t-1. In detail,

$$\psi_t(j,k) = \arg\max_{j-1\le i\le j} \left[\max_{0\le m\le \theta-1} \delta_{t-1}(i,m)\times a_{ij}\times r(t,j,k,m) \right] , \tag{20}$$

$$\rho_t(j,k) = \arg\max_{0\le m\le \theta-1} \left[\max_{j-1\le i\le j} \delta_{t-1}(i,m)\times a_{ij}\times r(t,j,k,m) \right] . \tag{21}$$

In terms of $\psi_t(j,k)$ and $\rho_t(j,k)$, the observation symbol sequence along the most probable path can then be backtracked as

$$s_n^* = NS , \qquad V_n^* = \arg\max_{0\le k\le \theta-1} \left[\delta_n(NS,k)\right] , \tag{22}$$

$$s_t^* = \psi_{t+1}(s_{t+1}^*,\ V_{t+1}^*) , \quad t = n\text{-}1,\ n\text{-}2,\ ...,\ 1 , \tag{23}$$

$$V_t^* = \rho_{t+1}(s_{t+1}^*,\ V_{t+1}^*) , \quad t = n\text{-}1,\ n\text{-}2,\ ...,\ 1 , \tag{24}$$

where n is the number of syllables in the sentence, NS is the number of states, s_t^* is the

state stayed and V_t^* is the pitch-contour VQ code selected for the lexical tone X_t on time t.

3.3 Pitch Frequency Interpolation

The pitch-contour VQ code for the t-th syllable is now known to be V_t^* according to Eq. (24). In terms of V_t^*, the time-normalized pitch-contour, *i.e.* 16 pitch frequency values, can be seen from the VQ codebook for the lexical tone X_t. Suppose that the pitch frequencies found are f_0, f_1, …, and f_{15} in Hz after scale conversion from logarithmic to linear, and the duration assigned to the voiced segment of the t-th syllable is DT in signal samples. Then, according to the definition of time-normalization, f_k is the pitch frequency on the k-th uniformly placed time point, T_k, and $T_k = DT * k / 15$. In terms of these pairs, (f_0, T_0), (f_1, T_1), …, (f_{15}, T_{15}), the pitch frequency on any given time point T can be interpolated using the closest pairs around T. Suppose that T is located between T_j and T_{j+1}. Then, the four pairs, (f_{j-1}, T_{j-1}), (f_j, T_j), (f_{j+1}, T_{j+1}), and (f_{j+2}, T_{j+2}), will be used to interpolate the pitch frequency on the time point T. Here, the method of Lagrange interpolation [25] is adopted and the formula is of the form in Eq. (1). In the case where j-1 is less than zero, the pair, (f_{j+3}, T_{j+3}), will be used instead. Similarly, if j+2 is greater than 15, the pair, (f_{j-2}, T_{j-2}), will be used instead.

3.4 A Variant Method for Applying SPC-HMM

Note that the prosodic information, breath-breaks and word-boundaries, are not used in training the SPC-HMM. Also, such information is not used in Section 3.1 to find the most probable path. Thus, it may be unclear whether the syllable pitch-contours generated according to the most probable path found in Section 3.1 are perceptually satisfactory. To study this problem, a second path-finding method that makes use of breath-break and word-boundary information is proposed and tested for applying the SPC-HMM. In this paper, the path-finding operating mode, described in Section 3.1, is called the Mode-A generation mode, and the path-finding operating mode that utilizes the prosodic information is called the Mode-B generation mode. In the generation mode, Mode-B, breath-breaks and word-boundaries are determined first by the text-analysis component. Then, such information is used to set up a state transition sequence for the SPC-HMM. Even though the state transition sequence is determined beforehand, finding the best observation symbol sequence for a sentence is still a two-dimensional (observation-symbol candidate axis and syllable-time axis) searching problem. The most probable path in the two-dimensional search space can be found by a conventional DP algorithm [20].

In the generation mode, Mode-B, the SPC-HMM used must be of exactly 3 states, and the state that a syllable would stay at is assigned according to some rules designed here. We will explain the rules through an example. Suppose that there is a breath-break between the third and forth syllables of a sentence consisting of seven syllables. Then, the state transition sequence will be set to 0, 1, 2, 1, 1, 2, and 2. The transition from state 2 to 1 is allowed in the Mode-B generation mode. That is, the state transition probability is changed to a value greater than 0 (*e.g.* 0.1). The first rule designed is: for the syllables within the first breath group, they are uniformly assigned to the 3 states. The second rule

designed is: for the syllables in the second or latter breath groups, they are uniformly assigned to the two states, 1 and 2. As to the word boundary information, they are used to adjust the state assignments made with the breath-breaks. The third rule designed is: the two syllables of a two-character word must be assigned the same state to have their pitch-contours generated by the same prosodic state. If this rule is violated, the second syllable's state is then reassigned to the first syllable's state. In addition, the fourth rule designed is: the last syllable in the first breath group must be assigned to State 1 or 2, and the last syllable in a latter breath group must be assigned to State 2. This rule is permitted to override the third rule. When these rules are followed, the state sequence will present a wave-like vibration in terms of the prosodic states, *i.e.* States 1 and 2 will be stayed alternatively.

4. SPC-HMM EXPERIMENTS AND PERCEPTION TESTS

4.1 Experiments for VQ Codebook Size

Since the SPC-HMM is a discrete HMM, a syllable's pitch-contour must be represented with a VQ code. Here, an interesting question is how many codewords a lexical tone's pitch-contour codebook should have. To answer this question, we conducted a series of testing experiments. In the first run, the first 30 sentences of the 375 recorded sentences were taken as the testing sentences, and the remaining sentences were used to train the SPC-HMM. In the second run, the second 30 sentences are taken as the testing sentences and the remaining sentences are used to train the SPC-HMM. Continuing in this manner, a total of seven runs of experiments were conducted. In addition, note that each run actually includes 10 experiments to test the five codebook sizes, *i.e.* 4, 6, 8, 10, and 12, under the two *NS* (number of states) values, 3 and 6, respectively. In detail, in each experiment, the steps indicated in Figs. 1 and 2 were executed in sequence. The steps include: (a) normalizing a pitch-contour's pitch-height, (b) training each lexical tone's codebook, (c) vector quantizing pitch-contours and encoding observation symbols, (d) training SPC-HMM, (e) generating pitch-contour VQ codes for the training sentences, *i.e.* inside test, and (f) generating pitch-contour VQ codes for the testing sentences, *i.e.* outside test. In Steps (e) and (f), the method used to generate a syllable's pitch contour is as described in Sections 3.1 and 3.2.

Here, the prediction error between a generated pitch contour and its corresponding recorded (already normalized on pitch-height) contour is measured with the formula given in Eq. (10). For each combination of a codebook size and one of the two *NS* values, prediction errors computed for the outside and inside tests are averaged. Then, the average errors are collected and averaged again for the seven runs. As a result, the average prediction errors for the inside and outside tests are as listed in Table 1 for different codebook sizes and *NS* values. For illustration, these error values are taken to draw the curves in Fig. 4.

Table 1. Pitch-contour prediction errors for 5 different codebook sizes.

Codebook size		4	6	8	10	12
NS=3	Inside	0.05015	0.05030	0.05014	0.04981	0.04950
	Outside	0.04994	0.05254	0.05300	0.05331	0.05250

NS=6	Inside	0.04466	0.04338	0.04321	0.04289	0.04279
	Outside	0.04593	0.04579	0.04579	0.04667	0.04665

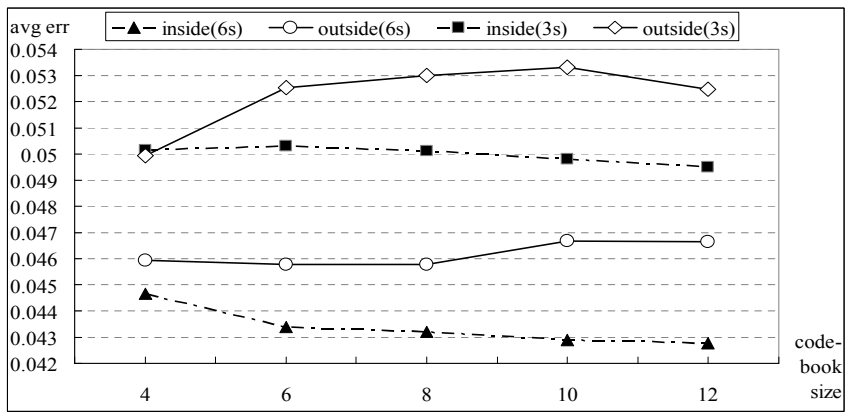

Fig. 4. Pitch-contour prediction error curves for 5 codebook sizes.

From Fig. 4, it can be seen that the two prediction error curves for the inside tests will both go downward as the codebook become larger. Nevertheless, this trend seems to be saturated for the lower curve obtained under the setting *NS*=6. In contrast, the two prediction error curves for the outside tests present different trends. The upper one, obtained under *NS*=3, goes upward and then downward while the lower one, obtained under *NS*=6, goes slightly downward and then upward. Also, it is apparent that the two curves obtained under *NS*=6 are both lower, *i.e.* of less prediction error, than the curves obtained under *NS*=3. Hence, according to the two curves obtained under *NS*=6, we think the codebook size "eight" is the best choice since the smallest outside test error and almost saturated inside test error will be obtained by using this size. Therefore, this codebook size is adopted in the following experiments.

4.2 Experiments for Number of States

Another question about using SPC-HMM is how many states an SPC-HMM should have. To study this problem, we fixed the size of the pitch-contour VQ codebook for each lexical tone to eight and varied the number of states (*NS*) from 3 to 7. Similarly, seven runs of experiments, as described in Section 4.1, were set up and conducted. Here, each run included five experiments to test the five *NS* values, *i.e.* 3, 4, 5, 6, and 7, respectively. For each *NS* value, prediction errors computed for the outside and inside tests were averaged. Then, the average errors were collected and averaged again for the seven runs. Basically, in each experiment, the steps indicated in Figs. 1 and 2 were executed in sequence. Additionally, we wanted to study the influence of pitch-height normalization here. Therefore, another five experiments were also set up and conducted in each run where the pitch-heights of the recorded syllables were not normalized. That is, the step of pitch-height normalization was skipped, and the recorded pitch-contours (without

pitch-height normalization) were used instead for measuring prediction errors. As a result, the average prediction errors for the inside and outside tests under the two conditions, with and without pitch-height normalization, were as those listed in Table 2 for different *NS* values. For illustration, these error values are taken to draw the curves in Figs. 5 and 6 for with and without pitch-height normalization, respectively.

Table 2. Pitch-contour prediction errors for 5 different *NS* values.

NS value		3	4	5	6	7
Pitch-height Normalized	Inside	0.05014	0.04722	0.04490	0.04322	0.04204
	Outside	0.05300	0.04957	0.04670	0.04578	0.04560
Pitch-height Un-normalized	Inside	0.07614	0.07546	0.07334	0.07171	0.07205
	Outside	0.09851	0.09463	0.09263	0.09244	0.09543

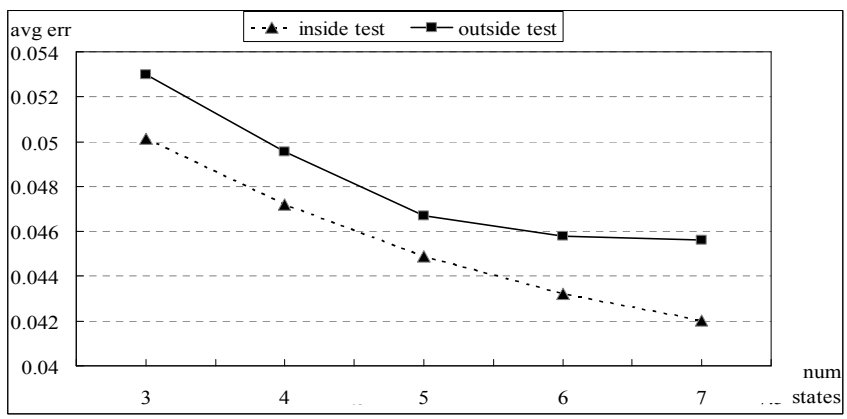

Fig. 5. Pitch-contour prediction error curves for 5 *NS* values with pitch normalization.

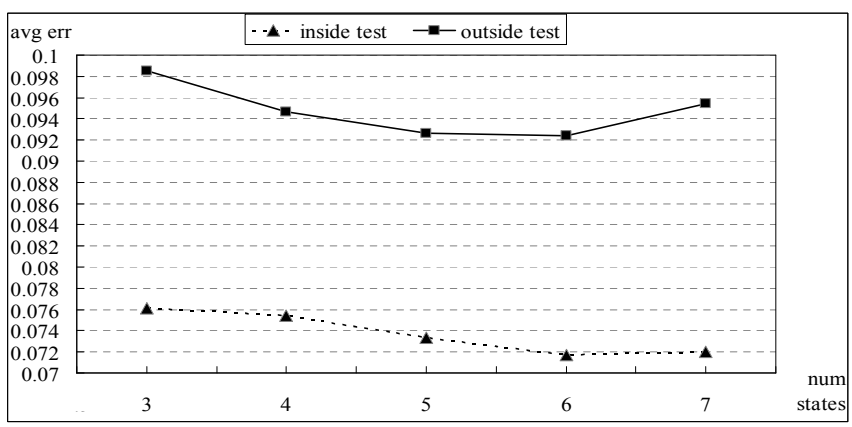

Fig. 6. Pitch-contour prediction error curves for 5 *NS* values without pitch normalization.

When the two figures, Figs. 5 and 6, are compared, it can be seen that the two curves

in Fig. 5 display much lower prediction errors than each of the curves in Fig. 6. Also, the gap between the two curves, for inside and outside tests, respectively, in Fig. 6 would be largely reduced to the gap between the two curves in Fig. 5. Therefore, the proposed pitch-height normalization procedures, as explained in Section 2.2, can indeed help reduce the prediction error of a generated pitch-contour by an SPC-HMM. In addition to the benefit observed for pitch-height normalization, another noticeable phenomenon can be found from both Fig. 5 and Fig. 6. That is, for outside tests, the prediction errors of the generated pitch-contours apparently will decrease as the number of states, *NS*, is increased. Nevertheless, the decreasing of prediction error would be saturated, as seen in Fig. 5, when *NS* becomes greater than 5. In contrast, the prediction error would be inversely increased, as seen in Fig. 6, when *NS* is set to 7. According to the trends of the curves for the outside tests, we chose to set *NS* to 6 for conducting perception tests.

4.3 Perception Tests: Intra-techniques

To perceptually evaluate the pitch-contour generation method proposed here, we built a prototype system to synthesize Mandarin speech. In this system, a word dictionary is referred to for parsing an input sentence into a sequence of words and for obtaining a word's pronunciation syllables. As to the placement of breath-breaks, an automatic method has not been developed yet. Therefore, each breath-break is indicated by the special character, "*", and is manually inserted into the sentence to be synthesized. As for the values of the prosodic parameters, syllable duration and amplitude, they are determined separately by two ANNs that were constructed in a previous study [2]. For signal synthesis, the synthesis method of HNMES [5] is adopted. Note that Mandarin has only 408 unique syllables if the lexical tones are not distinguished. Therefore, each of the 408 syllables is just recorded and saved once for analyzing its HNM parameters. Then, the same HNM parameters analyzed from a syllable are used to synthesize various syllable signals for different requested combinations of syllable durations and pitch-contours [5].

Utilizing the prototype system and the same article of 131 Chinese characters for input for each file, 3 speech files were synthesized via SPC-HMM under different parameter settings and generation modes. The 3 synthetic speech files are denoted here as SA, SB, and SC. SA was synthesized under the condition that the number of states and codebook size were set to 3 and 8, respectively, along with the Mode-A generation mode mentioned in Section 3.4 being adopted. To study the influence of the number of states, SB was synthesized by setting the value of *NS* to 6 while the codebook size and generation mode were kept the same as SA. In addition, to study the performance of the generation mode, Mode-B, SC was synthesized by setting the number of states and codebook size to 3 and 8, respectively, as set for SA, but the generation mode, Mode-B, was adopted instead. These 3 synthetic speech files can be accessed at http://guhy.csie.ntust.edu.tw/PitchCntr/SPCHMM.html .

After preparing the synthetic speech files, we invited 15 persons to participate in the perception tests. In the first run of tests, the speech files, SA and SB, were played in order to each of the participants. Then, he (or she) was requested to give a score to indicate which pitch-contour was more natural and preferred. The scores defined here are from -3 to 3. Among the seven scores, 3 (-3) means SB is much better (worse) than SA, 2 (-2) means SB is better (worse) than SA, 1 (-1) means SB is slightly better (worse) than SA, and 0 means SA and SB are not distinguishable. In the second run of tests, the speech

files, SB and SC, were played to each of the participants. Then, he (or she) was similarly requested to give a score to indicate which pitch-contour was more natural and preferred. After the perception tests, the scores given by the participants were collected for the two runs separately then averaged. The average scores obtained for the two runs are 1.533 and 0.467, respectively, and their standard deviations are 1.360 and 1.628, respectively.

According to the first score, 1.533, we think that the naturalness level of the generated pitch-contours with an SPC-HMM of 6 states is perceptually verified to be significantly better than the pitch-contours generated with an SPC-HMM of 3 states. On the other hand, according to the score, 0.467, we think that the generation mode, Mode-B, seems to be slightly better than (or comparable with) the generation mode, Mode-A, under *NS*=6 although an SPC-HMM of only 3 states is used here to operate the Mode-B generation mode.

4.4 Perception Test: Inter-techniques

Several ANN based pitch-contour generation methods have been proposed [13, 14], including a generation method we have already studied [29]. One may wonder whether the SPC-HMM based method can outperform an ANN based method. Therefore, we used the same recorded sentences to train and test the ANN designed in our previous study [29]. The structure of the ANN is illustrated in Fig. 7. That is, the ANN has 28 nodes in the input layer for inputting 8 contextual parameters, and 16 nodes in the output layer for outputting and representing a syllable's pitch contour. The 8 contextual parameters adopted are shown in Table 3. More details are given in our previous work [29]. In addition, the ANN has one hidden layer and one recurrent hidden layer. The number of nodes to be placed in the hidden layers was tested from 19 to 23. The best choice was found to be 21. When using 21 nodes in the hidden layers, we obtained average pitch-contour prediction errors of 0.03295 and 0.03619 for the inside and outside tests, respectively. The value, 0.03619, is apparently much better than 0.04560, the average error obtained from using SPC-HMM in outside tests. Therefore, according to objective measuring of pitch-contour prediction errors, the ANN based method will perform better than the SPC-HMM based method.

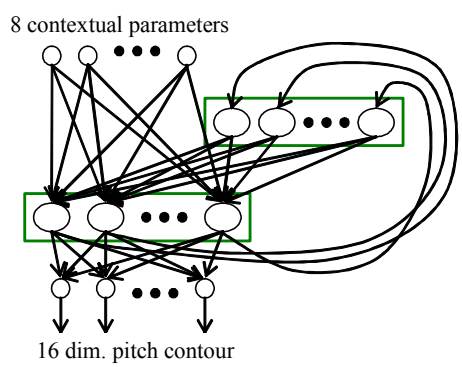

Fig. 7. Structure of our ANN for pitch contour generation.

Table 3. The 8 contextual parameters.

Items	Tone of previous syllable	Final class of previous syllable	Tone of current syllable	Initial of current syllable	Final of current syllable	Tone of next syllable	Initial class of next syllable	Time progress index
Bits	3	4	3	5	6	3	3	void

On the other hand, subjective perception tests were also conducted for comparing the SPC-HMM based method with the ANN based method. For the SPC-HMM based method, the synthetic speech file, SC, as described in Section 4.3, was selected as the representative. As for the ANN based method, the same speech synthesis system was used, except that the pitch contour generation module was replaced with the ANN module using 21 nodes in the hidden layers. Here, the speech file synthesized using the ANN pitch contour generation module is denoted as SD. In terms of SC and SD, each of the 15 invited persons was requested to give a score after he (or she) listened to the two files. A score of positive value means SD is better and preferred over SC in naturalness level of synthetic pitch contours. In contrast, a score of negative value means SC is better and preferred. The detailed definitions for the allowed scores are the same as those given in Section 4.3. After perception tests, the scores given by the 15 persons were collected and averaged. The average score obtained is -0.600, and its standard deviation is 0.952. Therefore, the naturalness level of the SPC-HMM based method is slightly better than (or comparable to) that of the ANN based method. One explanation for such perception-test result is that the information of prosodic states (represented as the states of an HMM) and word boundaries are explicitly utilized in the SPC-HMM based method. In contrast, the information of word boundaries is not used by the ANN model, and the transitions between prosodic states are not explicitly knowable for the ANN model. When the results of the two evaluations (objective prediction-error measuring and subjective perception testing) are put together, we think that a variant speaking style realized via the SPC-HMM generated syllable pitch-contours may be perceived as natural and preferred although the pitch-contours are not like those originally uttered.

5. CONCLUSIONS

In this paper, we have studied and proposed a syllable pitch-contour generation method that uses discrete HMM to model the implicit prosodic states stayed and that encodes a discrete observation symbol in terms of vector quantizing a syllable's pitch-contour. In this method, SPC-HMM, the criterion of sentence-wide optimization is used to define the probability of a syllable pitch-contour VQ code sequence. To find the sequence of the highest probability efficiently, we have also developed a dynamic programming based algorithm. In addition, we have proposed a simple but effective method for pitch-height normalization. Via this method, abnormal pitch-contour transitions between adjacent syllables almost can be eliminated in synthesized speech.

For the size of a lexical tone's pitch-contour codebook, we have conducted a series of testing experiments. The results of the experiments show that using a codebook of 8 codewords is the best choice. As to the number of states for structuring an SPC-HMM,

our experimental results show that an SPC-HMM of 6 states will generate pitch-contours with saturated prediction errors. In addition, according to the perception tests' results, an SPC-HMM of 6 states can indeed outperform an SPC-HMM of 3 states if they are operated in the Mode-A generation mode. Nevertheless, another result is still notable. In the Mode-B generation mode, the prosodic information, breath-breaks and word boundaries, are utilized to set the state staying sequence for an SPC-HMM to generate syllable pitch-contours. After perceptual testing, it is found that an SPC-HMM of 3 states operated in the Mode-B mode can generate syllable pitch-contours that are slightly more natural than those generated by an SPC-HMM of 6 states operated in the Mode-A mode.

REFERENCES

1. C. Shih and R. Sproat, "Issues in text-to-speech conversion for Mandarin," *Int. Journal of Computational Linguistics and Chinese Language Processing*, Vol. 1, 1996, pp. 37-86.
2. H. Y. Gu and C. Y. Wu, "Model spectrum-progression with DTW and ANN for speech synthesis," in *Proc. ECTI-CON 2009*, Pattaya, Thailand, 2009, pp. 1010-1013.
3. K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to English," *IEEE Workshop on Speech Synthesis*, Santa Monica, CA, 2002, pp. 227-230.
4. H. Y. Gu and W. L. Shiu, "A Mandarin-syllable signal synthesis method with increased flexibility in duration, tone and timbre control," *Proc. Natl. Sci. Counc. ROC(A)*, Vol. 22, 1998, pp. 385-395.
5. H. Y. Gu and Y. Z. Zhou, "An HNM based scheme for synthesizing Mandarin syllable signal," *Int. Journal of Computational Linguistics and Chinese Language Processing*, Vol. 13, 2008, pp. 327-341.
6. E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, Vol. 9, 1990, pp. 453-467.
7. Y. Stylianou, "Modeling speech based on harmonic plus noise models," *Nonlinear Speech Modeling and Applications*, Editors G. Chollet, *et al.*, Springer-Verlag, Germany, 2005, pp. 244-260.
8. R. Sproat, *Multilingual Text-to-speech synthesis: The Bell Lab's Approach*, Kluwer Academic Publishers, Boston, 1998.
9. H. Fujisaki, "Dynamic characteristics of voice fundamental frequency in speech and singing," *The Production of Speech*, P. F. MacNeilage ed., Springer-Verlag, New York, 1983, pp. 39-55.
10. L. S. Lee, C. Y. Tseng, and C. J. Hsieh, "Improved tone concatenation rules in a formant-based Chinese text-to-speech system," *IEEE trans. Speech and Audio Processing*, Vol. 1, 1993, pp. 287-294.
11. M. S. Yu, N. H. Pan, and M. J. Wu, "A statistical model with hierarchical structure for predicting prosody in a Mandarin text-to-speech system," *Int. Symposium on Chinese Spoken Language Processing,* Taipei, Taiwan, 2002, pp. 21-24.
12. M. Dong, K. T. Lua, "Pitch contour model for Chinese text-to-speech using CART and statistical model," *Int. Conference on Spoken Language Processing,* Denver,

USA, 2002, pp. 2405-2408.
13. S. H. Chen, S. H. Hwang, and Y. R. Wang, "An RNN-based prosodic information synthesizer for Mandarin text-to-speech," *IEEE trans. Speech and Audio Processing*, Vol. 6, 1998, pp. 226-239.
14. C. T. Lin, R. C. Wu, J. Y. Chang, and S. F. Liang, "A novel prosodic-information synthesizer based on recurrent fuzzy neural network for the Chinese TTS system", *IEEE trans. Systems, Man, and Cybernetics*, Vol. 34, 2004, pp. 309-324.
15. W. H. Lai, *A Statistic Prosodic Modeling Modeling for Mandarin Speech*, Ph.D. Dissertation, Institute of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, 2003.
16. A. Ljolje and F. Fallside, "Synthesis of natural sounding pitch contours in isolated utterances using hidden Markov models," *IEEE trans. Acoust., Speech and Signal Processing*, Vol. 34, 1986, pp. 1074-1080.
17. T. Fukada, Y. Komori, T. Aso, and Y. Ohora, "A study on pitch pattern generation using HMM-based statistical Information," *Int. Conference on Spoken Language Processing*, Yokohama, Japan, 1994, pp. 723-726.
18. H. Y. Gu and C. C. Yang, "A sentence-pitch-contour generation method using VQ/HMM for Mandarin text-to-speech", *Int. Symposium on Chinese Spoken Language Processing*, Beijing, China, 2000, pp. 125-128.
19. D. O`Shaughnessy, *Speech Communication: Human and Machine*, second ed., IEEE Press, Piscataway, New Jersey, 2000.
20. L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
21. S. H. Chen and Y. R. Wang, "Vector quantization of pitch information in Mandarin speech," *IEEE trans. Communications*, Vol. 38, 1990, pp. 1317-1320.
22. J. C. Chen, J. S. R. Jang, and T. L. Tsai, "Automatic pronunciation assessment for Mandarin Chinese: approaches and system overview," *Int. Journal of Computational Linguistics and Chinese Language Processing*, Vol. 12, 2007, pp. 443-458.
23. J. Teutenberg, C. Watson, and P. Riddle, "Modeling and synthesizing F0 contours with the discrete cosine transform," *Int. Conf. Acoust., Speech and Signal Processing*, Las Vegas, USA, 2008, pp. 3973-3976.
24. S. Ravuri and D. P. W. Ellis, "Stylization of pitch with syllable-based linear segments", *Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, USA, 2008, pp. 3985-3988.
25. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, second ed., Springer-Verlag, New York, 1993.
26. W. J. Yang, J. C. Lee, Y. C. Chang and H. C. Wang, "Hidden Markov model for Mandarin lexical tone recognition," *IEEE trans. Acoust., Speech and Signal Processing*, Vol. 36, 1988, pp. 988-992.
27. K. Sayood, *Introduction to Data Compression*, second ed., Morgan Kaufmann, San Francisco, CA, 2000.
28. K. Sugawara, *et al.*, "Isolated word recognition using hidden Markov models," *Int. Conf. Acoustics, Speech, and Signal Processing*, Tampa, Florida, USA, 1985, pp. 1-4.
29. H. Y. Gu, Y. Z. Zhou, and H. L. Liau, "A system framework for integrated synthesis of Mandarin, Min-nan, and Hakka speech," *Int. Journal of Computational Linguistics and Chinese Language Processing*, Vol. 12, 2007, pp. 371-390.