Chapter 3. Preprocessing of Corpus-based TTS/SVS

As described in Chapter 1, phonetic segmentation is an important task in speech/singing voice synthesis since the higher segmentation accuracy implies that the better synthetic quality can be achieved. In addition to the phonetic segmentation, other essential tasks, such as corpus design/collection, phonetic transcription, pitch estimation, and pitch marking, are also very important to speech/singing voice synthesis. They shall be addressed in this Chapter.

3.1. Corpus Design Principle

Since the size of the corpus is not infinite, careful design of the corpus becomes very significant for corpus-based TTS/SVS systems. Two primary concerns of the corpus design are “phonetically rich” and “prosodically rich”. In general, the goal of a “phonetically rich” corpus is to cover various phonetic combinations as many as possible in a corpus of acceptable size. Here phonetic combinations include intra-syllabic and inter-syllabic structures. In other words, the phonetically rich corpus should cover all possible INITIAL-FINAL, FINAL-INITIAL, and FINAL-FINAL structures for Mandarin Chinese (Here INITIAL represents the consonant and FINAL means the vowel or diphthong). In addition, since Mandarin is a tonal language, it should be necessary to consider the syllable-tone combinations further. On the other hand, to design a “prosodically rich” corpus is to cover as many different prosodic behavior of Mandarin as possible. In other words, the modalities of utterances, such as interrogative, exclamatory and declarative, need to be considered. Here part-of-speech, punctuation marks and bigram/trigram

coverage are used to identify these modalities. More detailed descriptions are referred to the literature [3]. The following diagram shows an example of the corpus design. This corpus was used for our anger-based emotional transformation model TTS system [41]:

1. *Initially, set the score as 5 for each anger-related keyword and to 2 for each Chinese tonal syllable.*
2. *Calculate the score of each sentence as the total score for distinct syllables and distinct anger-related keywords appearing in this sentence. Normalize the score for the length of the sentence.*
3. *Select the sentence with the highest score for recording.*
4. *Decrease the scores of the syllables and anger-related keywords within the just selected sentence by one (till zero) to reflect their lessening importance in the next iteration.*
5. *Repeat steps 2 to 4 until 1000 sentences are collected.*

The goal of the design method is to collect top 1000 “anger-suitable” sentences to be recorded by ranking about 10 000 anger-based sentences. The principle is to cover as many as possible of the Chinese tonal syllables together with anger-related keywords.

3.2. Phonetic Transcription

After the corpus design/collection, the subsequent task is to label the phonetic transcription. However, this is not an easy task because some Chinese characters have multiple syllables with different pronunciations depending on their lexical contexts. For example, the Chinese character 『著』 is pronounced “ㄓㄨˋ” (“zhu4”) in『著名』

(meaning “famous”) and “ㄓㄠˊ” (“zhao2”) in『睡著』 (meaning “sleep”).

Generally speaking, there are two common ways to design automatic phonetic transcription. One is to apply word segmentation in Chinese natural language processing (NLP), which can be implemented via several methods, such forward or backward maximum word matching algorithm [42][43], dynamic-programming-based statistic probability method [44], etc. The other one is to employ forced alignment of the Viterbi search based on HMM.

As a matter of fact, word segmentation methods do not necessarily outperform forced alignment method, and vice versa. For instance, word segmentation relies on a collection of Chinese words in the form of a dictionary, which cannot cover all existing words since new words are constantly being created. In addition, it is somewhat difficult to distinguish some similar pronunciations via forced alignment. For example, “便” is pronounced “ㄆㄧㄢˊ” (“pian2”) in “便宜” (“meaning cheap”) and is pronounced “ㄅㄧㄢˋ” (“bian4”) in “方便” (“meaning convenient”). In view of this, to avoid errors resulting from automatic phonetic transcription, a better way is to combine word segmentation and forced alignment together through the following two steps:

1) Word segmentation is performed by using both forward and backward maximum matching based on a word dictionary. Then keep the phonetic transcriptions as candidates for use in the next step. (If the result is the same, only a single phonetic transcription is kept.)
2) These different phonetic transcription candidates are evaluated by using a forced alignment through Viterbi decoding. The phonetic transcription that has the

maximum log likelihood is accepted finally.

The above two steps combine both word segmentation in NLP and forced alignment in speech recognition to achieve better phonetic transcription performance. Based on the experimental result of our prior work [45], the syllable error rate was about 2% while using the step 1 alone, whereas the error rate was reduced to 1% or so while using both the two steps. In other words, a significant reduction of 50% in the error rate could be achieved.

3.3. Pitch Estimation/Marking

3.3.1. Pitch Estimation

Various pitch estimation methods have been proposed in the literature, such as the autocorrelation factor [8], pattern recognition [9], least-square fitting [10], dynamic programming based (DP-based) algorithm [46], etc. In this dissertation, we used the DP-based method to calculate the pitch information via Speech Filing System software [47]. This is due to the fact that the DP-based method is able to calculate the pitch contours of utterances accurately in most cases. However, a small amount of error results estimated by the DP-based method are still in need of manual revision further.

3.3.2. Pitch Marking

TD-PSOLA [11] is the most widely used method for pitch/duration modification in concatenation-based speech synthesis. An essential part of TD-PSOLA is pitch marking, which tries to find the glottal closure instant (GCI) in order to perform synchronous analysis. If the result of pitch marking is not good enough, the synthesized voices generated by using TD-PSOLA will possibly have buzzy effects. As documented in Chapter 1, several

approaches [14][48][49] have been proposed to tackle the pitch marking problem. In one of our prior work [15], we have proposed a two-phase algorithm for detecting pitch marks in a reliable manner.

In phase 1, we need to decide to use pitch marks at peaks (local maxima) or valleys (local minima). This is motivated by our experiments which indicate that, in most cases, valley-based pitch marks are more robust than peak-based ones. Consequently, we employ a quick scheme for choosing peaks or valleys as our candidates for a detailed pitch marking method in phase 2.

In phase 2, we use dynamic programming to extract optimal pitch marks. Our computation is based on speech waveform only, so it is more efficient than wavelet-based approach [49]. The proposed approach is similar to Viterbi decoding [50]. The key point is to define state and transition scores, which is explained later. Fig. 3.1 shows the processing of the two-phase algorithm for detecting pitch marks.

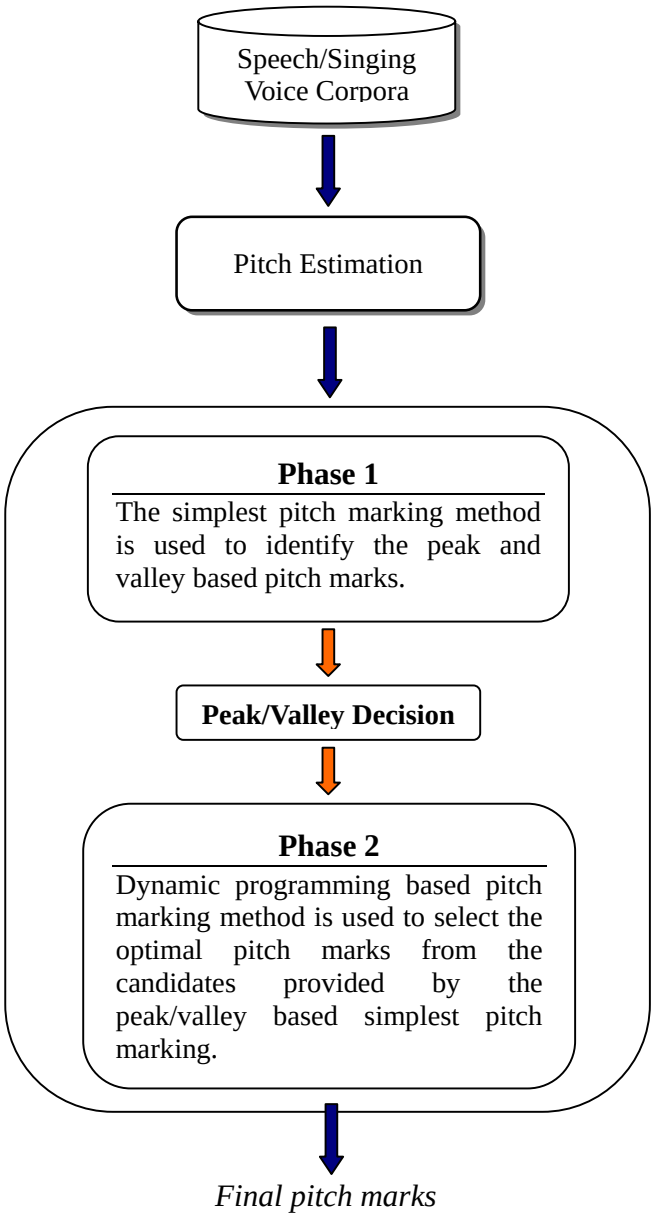

Fig. 3.1. The processing of the proposed two-phase algorithm for detecting pitch marks.

Before we elaborate the two-phase algorithm, we briefly introduce the simplest pitch marking algorithm as follows.

1) Find the global maximum of the waveform. Denote its time coordinate as t_m. This is the first pitch mark.
2) Search other pitch marks to the right of t_m by finding the maximum in the search region $[t_m + f*T_0, t_m + (2-f)*T_0]$, where T_0 is the pitch period and f is factor whose range could be 0.5~0.9. Repeat the same procedure until all pitch marks (maximum points) to the right of the global maximum are found.
3) Repeat step 3 to find pitch marks to the left of t_m. (The search region should be $[t_m - f*T_0, t_m - (2-f)*T_0]$ instead.)

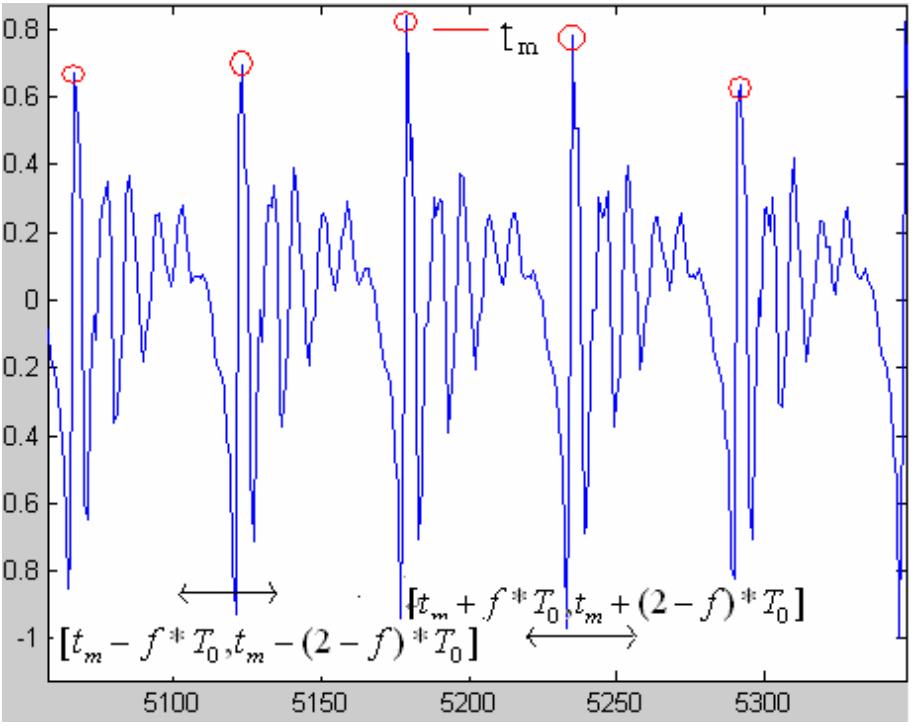

Fig. 3.2. An example of the simplest pitch marking method.

The above algorithm can identify peak-based pitch marks. A similar method can be easily conceived to find valley-based pitch marks. There is no denying that the two types of pitch

marks do not necessarily have the same performance even though they are identified under the similar pitch marking method. As a consequence, we need to apply a reasonable scheme to make peak/valley decision before using a sophisticated method for pitch marking. Here we propose a simple method listed as follows to make good decision between peak and valley based pitch marks:

1) Perform simplest pitch marking method to find the peak and valley based pitch marks. Denote the corresponding instantaneous pitch curves as P_p and P_v, respectively.
2) Find the pitch contour from each frame and interpolate the pitch contour to have the same length as P_p and P_v. Make peak/valley decision based on their similarity to the interpolated (or "roughly correct") pitch curves.

Fig. 3.3 shows the result of peak-based pitch marking. Its deviation from the interpolated pitch curve is quite obvious. (The pronunciation of this speech waveform is "ㄖㄜ" ("re").) On the other hand, Fig. 3.4 shows the result of valley-based pitch marking which is better with less deviation.

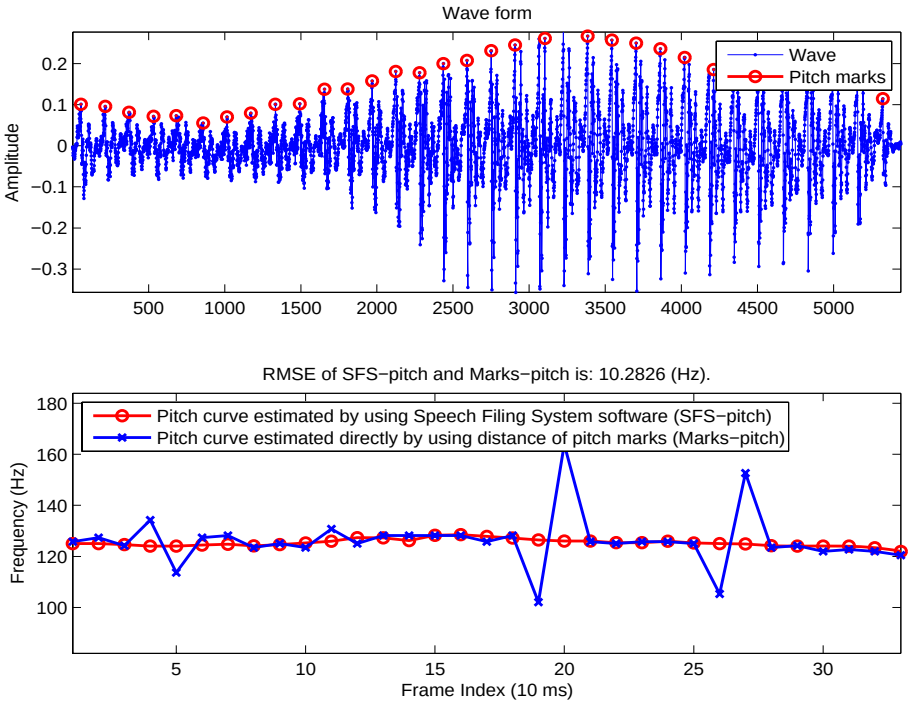

Fig. 3.3. The result from peak searching based pitch marking.

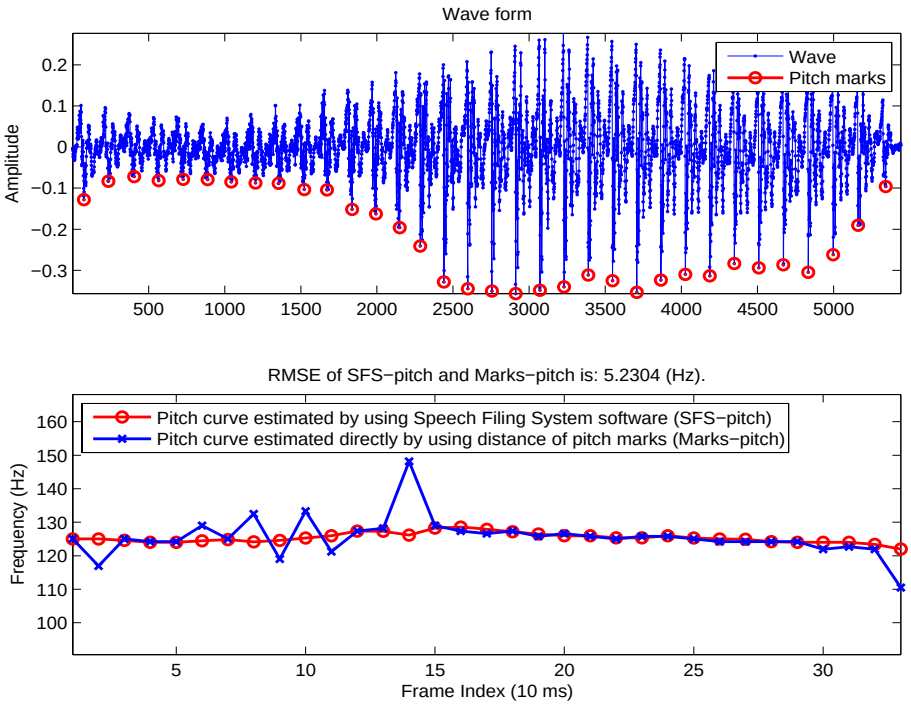

Fig. 3.4. The result from valley searching based pitch marking.

In the above example, the valley-based pitch marks seem better than the peak-based pitch marks according to the peak/valley decision rules mentioned above. Hence we subsequently perform the DP-based pitch marking based on the valley-based results. For simplicity in the later discussion, we shall assume that the peaks have been picked as candidates for pitch marks. If valleys are picked, we can simply flip the signals up side down and follow the same discussion.

Generally speaking, good candidates of pitch marks should have the following two characteristics:

1) Their magnitude should be large.
2) Their distances to the next adjacent pitch marks represent instantaneous pitch. Since pitch should be a smooth function, the distance (as a function of the pitch mark index) should also be smooth.

However, these two criteria are not always in agreement. As a result, we propose a method based on dynamic programming to find the best pitch marks. Specifically, we shall preserve K candidates in each search region instead of only one pitch mark in the aforementioned simplest pitch marking method. In this dissertation, the value of K is set to 3 for achieving the optimal tradeoff between performance and efficiency. Since the above two criteria are sometimes contradictory, we convert the criteria into scores and then try to find the optimum pitch marks with the maximal accumulated score. The approach to find the

optimal pitch marks is close to Viterbi search, in which dynamic programming is used to find the optimal path in an efficient manner. This is achieved via the following three steps.

1) Compute the state score of each candidate in a search region. The state score of a candidate (peak) is roughly proportional to its height:

$$S_i(j) = \frac{h_i(j) - h_{\min}}{h_{\max} - h_{\min}}, \quad j = 1,..K. \quad (3.1)$$

Each search region contains at most K pitch-mark candidates. The state score associated with candidate j in search region i should be related to the height of candidate j. In other words, the higher the candidate is, the more likely it is a pitch mark. Here $h_i(j)$ denotes the height of candidate j in region i; and $h_{\max}$ and $h_{\min}$ are the maximum and minimum of the voice segment, respectively.

2) Compute the transition score of each candidate in a search region. The transition score depends on how close the distance to the average pitch period of the entire voice segment:

$$T_i(j_1, j_2) = \frac{1}{1 + \frac{\left| F - \frac{f_s}{d} \right|}{F}}, \quad (3.2)$$

where F is the "most likely" pitch frequency at the middle between search regions i and $i+1$. This is achieved by interpolating the pitch curve obtained from each frame using the common practice of pitch tracking. Besides, f_s is the sampling rate and d is the distance (in terms of sample points) between candidate j_1 and j_2.

3) Apply dynamic programming to find the up-to-now optimal accumulated score until all candidates are processed. Then back track to find those pitch marks that contribute to the optimal accumulated score.

In a preliminary experiment, we used the same speech waveform that was adopted in the first phase to examine the performance of this scheme. Fig. 3.5 demonstrates that the proposed two-phase pitch marking approach does outperform the simplest pitch marking method introduced earlier. Furthermore, we have conducted several experiments demonstrated in [15] to validate the efficiency and effectiveness of the proposed method. In fact, we utilized the pitch marks not only to modify the pitch/duration via TD-PSOLA but also to create a smooth transition between frames. In one of our prior work [51], we performed the frame concatenation based on synchronized pitch marks such that we can avoid producing undesirable buzzy sounds in the process of voice conversion.

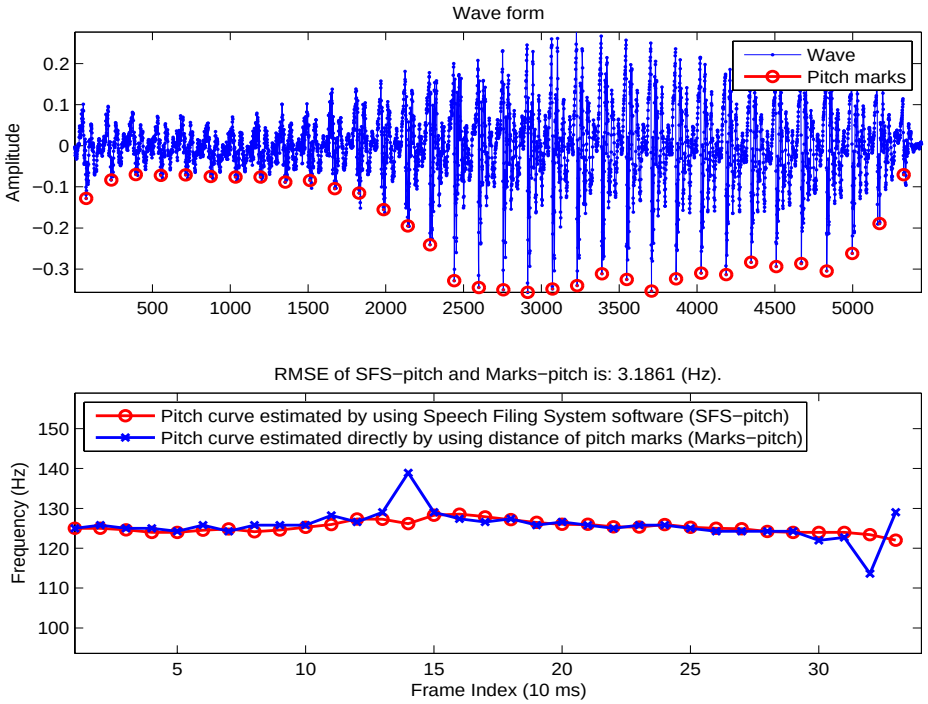

Fig. 3.5. The result of pitch marking based on dynamic programming.