Chapter 5. Boundary Refinement Based on Hybrid Approach

As documented in Chapter 4, we employ the HMM-based alignment to perform initial phonetic segmentation on speech data, whereas we adopt both the HMM-based and DTW-based alignments to carry out the same task on singing voice data. The experimental results revealed that the two kinds of initial estimates are not good enough in practice. As a result, we subsequently introduce a post-processing scheme to refine the initial segmental results. In this chapter, we propose an effective hybrid approach to perform the boundary refinement. The detailed descriptions of the proposed approach shall be elaborated in the following sections.

5.1. Phonetic Transition Categories in Mandarin

Mandarin Chinese is a tonal language in which each character is associated with a tone and a base syllable. About 1400 tonal syllables comprise the whole set of all legal combinations of 411 base syllables and five tones. Since syllables are generally regarded as the basic synthesis units in most Mandarin TTS/SVS systems, the goal of automatic phonetic segmentation is to precisely refine the boundaries between two consecutive syllables. In this study, the boundary refinement will be performed only on the boundary between two consecutive syllables.

There are 22 INITIAL and 38 FINAL in Mandarin Chinese. In order to avoid the problem of insufficient data coverage for training, six primary types for INITIAL and nine primary types for FINAL are defined in advance according to their similarity in articulation. Therefore, a total of 54 (9 x 6) models for these transitions are constructed. Table 5.1 and

Table 5.2 list the types of INITIAL and FINAL (in Hanyu Pinyin) used in this dissertation.

TABLE 5.1 SIX TYPES OF INITIAL

1	m, n, l, r, “null”	2	h, x, sh	3	b, d, g
4	j, zh, z	5	p, t, k	6	q, ch, c, f, s

TABLE 5.2 NINE TYPES OF FINAL

1	“null”	2	a, ya, wa	3	o, wo
4	e, er	5	ê, ye, yue	6	ai, iai, wai, yi, ei, wei
7	ao, yao, wu, yu, ou, you	8	an, yan, wan, yuan, en, yin, wen, yun	9	ang, yang, wang, eng, ying, weng, yong

5.2. Feature Definition

In order to refine the boundaries identified by the HMM-based or DTW-based recognizers, we need to employ several potential acoustic features in addition to MFCCs and pitch contours. Some of these acoustic features are commonly used in speech processing; they include the zero-crossing rate, log energy, line spectrum pairs (LSPs) [58]. In addition, we also added other two useful acoustic features, spectral entropy [59] and bisector frequency [45]. Here we briefly introduce the two acoustic features, entropy and bisector.

5.2.1. Entropy

For each frame, the spectrum is calculated through fast Fourier transform (FFT). This FFT spectrum can be regarded as a vector of coefficients in the orthonormal basis. The probability density function for the spectrum is estimated by normalization over all frequency components:

$$p_i = \frac{S(f_i)}{\sum_{k=1}^{N} S(f_k)}, i = 1...N \quad , \qquad (5.1)$$

where $S(f_i)$ is the spectral energy for the frequency component f_i, p_i is the

corresponding probability density, and *N* is the total number of frequency components in FFT spectrum. The spectral entropy for each frame is defined as:

$$SpectralEntropy = -\sum_{k=1}^{N} p_k \log p_k \quad . \quad (5.2)$$

5.2.2. Bisector Frequency

The bisector frequency is defined in equations (5.3) and (5.4):

$$FreqIndex = \arg\min_{1<J<N} \left| \sum_{i=1}^{J} S(f_i) - \frac{\sum_{i=1}^{N} S(f_i)}{2} \right|, \quad (5.3)$$

$$Bisector Freq = \frac{FreqIndex}{N} \times \frac{SampleRate}{2}, \quad (5.4)$$

where $S(f_i)$ is the spectral energy for the frequency component f_i and there are *N* distinct frequency components in the spectrum. The key characteristic of the bisector frequency is that its value is smaller for a voiced frame but larger for an unvoiced frame. Hence we can use this feature to distinguish unvoiced from voiced patterns. Although the zero-crossing rate could also be used to detect unvoiced patterns, it is not sufficiently robust, especially when the mean amplitude of an unvoiced frame deviates from zero [45]. In the implementation, we normalize the value of this feature to the range [0,1] according to equation (5.5):

$$Bisector Freq = \left(\frac{Bisector Freq - LowFreq}{HighFreq - LowFreq} \right) \quad (5.5)$$

where the values of *HighFreq* and *LowFreq* are empirically set to be $\frac{SampleRate}{2} \times 0.8$

Hz and 100 Hz, respectively.

Fig. 5.1 shows the spectral entropy and the bisector frequency of the sentence, “我明年將離開彰化去日本” (“uo3-ming2-nian2-jiang-li2-kai-xang-hua4-ju4-r4-ben3”).

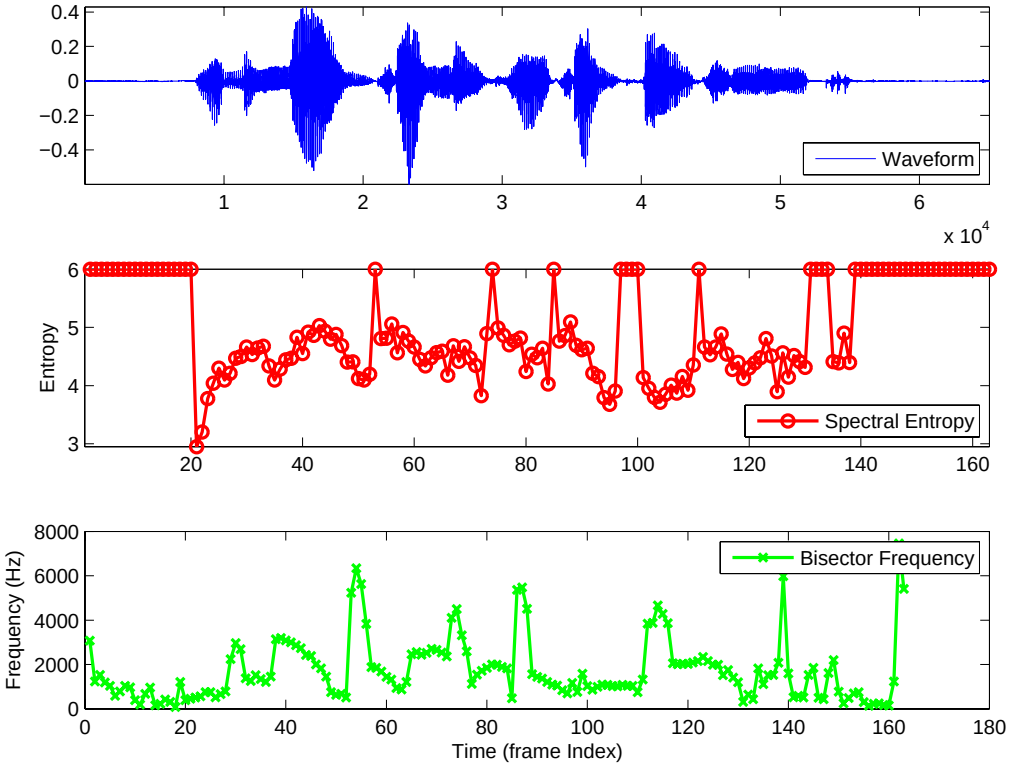


Fig. 5.1. The spectral entropy and the bisector frequency of the sentence, “我明年將離開彰化去日本” (“uo3-ming2-nian2-jiang-li2-kai-xang-hua4-ju4-r4-ben3”).

5.2.3. Acoustic Feature Vector

As mentioned previously, we have chosen the seven commonly used acoustic features as the basic feature set, including zero-crossing rate, log energy, LSFs, spectral entropy, bisector frequency, pitch contours, and MFCCs. These basic features are extracted from a frame centered at each candidate boundary. Since LSFs and MFCCs are both a

12-dimensional vector, the basic features consist of a 29-dimensional feature vector. Here the basic features are referred to as the static features for later discussions.

Generally speaking, using the static features alone to represent a boundary is probably insufficient. In most of the studies regarding speech/speaker recognition it is common to use the static features and their corresponding delta features, for instance, MFCCs and its delta MFCCs. As a result, in order to estimate the rate of change of the static features, their delta features can be computed via the delta function (5.6).

$$\Delta F(t) = \frac{\sum_{\tau=-M}^{M} F(t+\tau)\tau}{\sum_{\tau=-M}^{M} \tau^2}, \qquad (5.6)$$

where F denotes the static features, M is set to 2, t denotes a frame index, and ΔF denotes the corresponding delta features. In this study, the frame size is 20 ms and the frame overlap is 10 ms. For each candidate boundary, we can obtain a 58-dimensional feature vector including the static features and the delta features. Eventually, the 58-dimensional feature vector is further normalized to have zero sample mean and unity sample variance for each component. It is noted that we utilize the 58-dimensional normalized feature vector to perform phonetic segmentation in both speech and singing voices.

5.3. Candidate Boundaries for Training

Based on our observation, the accuracy of TTS-455 within ±50 ms segmentation error tolerance is approximately 96 percent using HMM-based initial segmentation. Accordingly, we collected a set of candidate boundaries located within ±50 ms of a true (manually labeled) boundary. Here we created a set of training data by adding several candidate

boundaries, 5 ms apart, located within ±50 ms of a true boundary. A candidate boundary is labeled “correct” if it is within ±10 ms of the true boundary. Chou *et al.* [3] indicated that manual labeling of two human experts can achieve about 90% consistency on 10 ms tolerance. Therefore we chose to have 5 “correct” candidates (including the true one), all within ±10 ms of the manually labeled one, for our experiments. On the other hand, we chose 6 “wrong” candidates located within [40, 50] ms and [-40,-50] ms of the true boundary. In other words, for each true boundary, we create a set of 11 candidate boundaries, which have 5 labeled “correct” and 6 labeled “wrong” as their desired classification output as shown in Fig. 5.2.

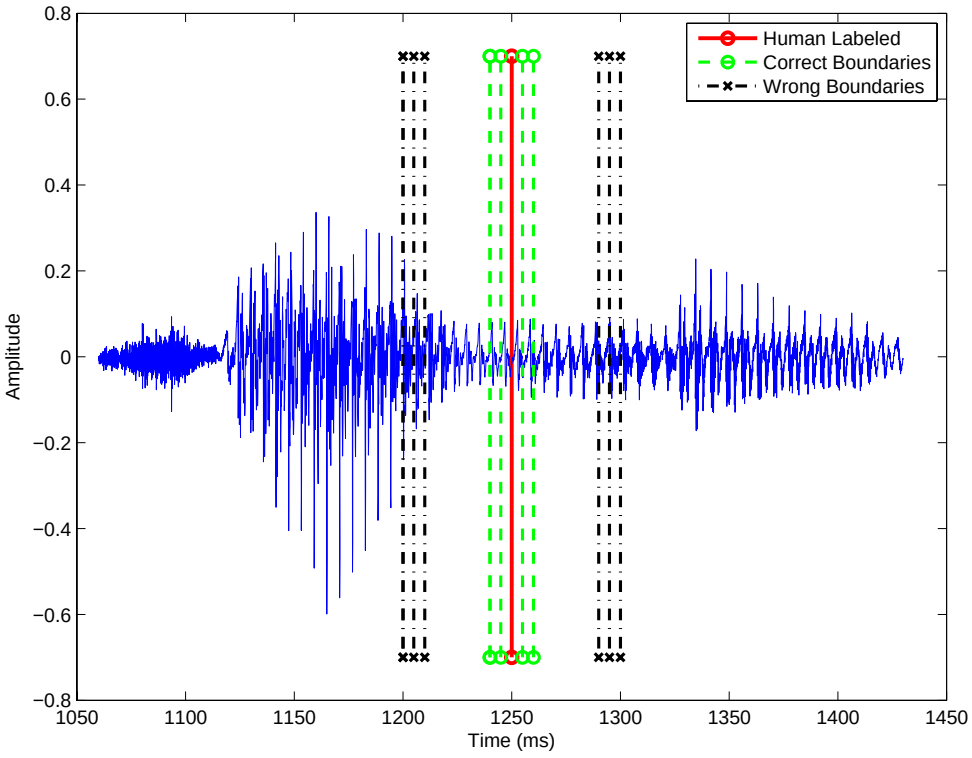
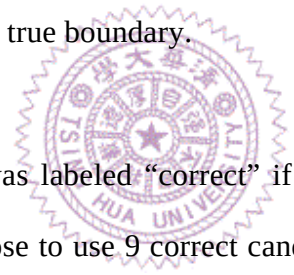


Fig. 5.2. Training data of 5 correct boundaries and 6 wrong boundaries around the true boundary labeled by humans. The content of this speech waveform was “將離” (“jiang-li2”).

For singing voice data, the 200 ms segmentation accuracy is close to 97% while applying either DTW-based or HMM-based alignment on SVS-1384. Thus, given a true boundary, the candidate boundaries located within ±200 ms of this boundary are collected to form the training data. More specifically, these candidate boundaries are collected using the following procedure:

1) Add a set of candidate boundaries 5 ms apart, located within ±20 ms of the true boundary.
2) Add a set of candidate boundaries 20 ms apart, located within 120 ~ 200 ms and -120 ~ -200 ms around the true boundary.



Here a candidate boundary was labeled "correct" if it was within ±20 ms of the true boundary. In other words, we chose to use 9 correct candidates in our experiments. On the other hand, we label the other candidate boundaries as "wrong" if it located within 120 ~ 200 ms and -120 ~ -200 ms around the true boundary. Eventually, we created a set of 19 candidate boundaries (including the true one), with 9 labeled "correct" and 10 labeled "wrong" as the desired classification output as shown in Fig. 5.3.

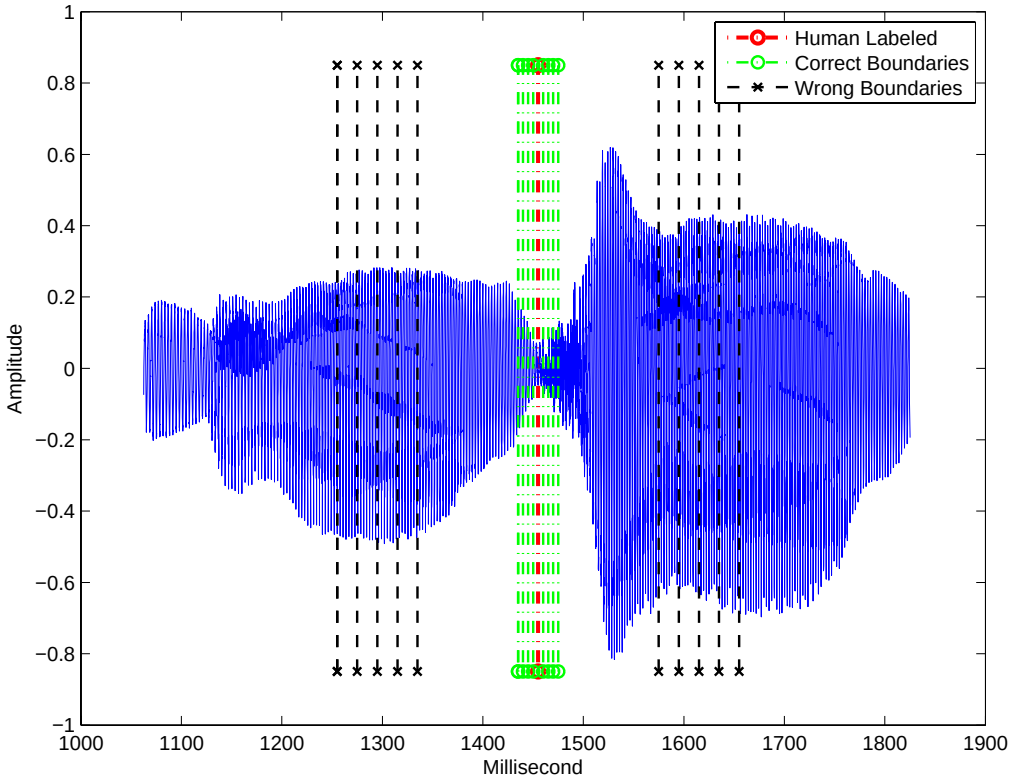


Fig. 5.3. Training data of 9 correct boundaries and 10 wrong boundaries around the true boundary labeled by humans. The content of this singing voice waveform was “寧靜” (“ning2-jing4”).

5.4. Statistics-based Method

Although we have defined acoustic features, it is possible that not all acoustic features work very well for each phonetic transition category. In order to achieve better performance, it is probably helpful if we consider feature selection further for each phonetic transition category. There are several feature selection methods for classification, such as sequential forward selection [60], sequential backward selection, sequential floating search method [61], etc. Generally speaking, the performance difference among these methods is usually not obvious. In this study we chose sequential forward selection (SFS) to proceed with later feature selection due to its efficiency. The principle behind SFS is to

start with a single feature with the best classification rate. Then we try to identify a newly added feature to the already selected feature set that can most increase the classification rate. This greedy step is repeated until the desired number of features has been selected, or until there is no improvement in the classification rate.

Once these boundaries with its corresponding acoustic features are collected, we subsequently construct its corresponding classifier which is used to classify the input candidate boundary into “correct” or “wrong” category. There are several common methods for the construction of a classifier, such as k-nearest neighbor rule (KNNR) [62], Gaussian mixture model (GMM) [63], neural network (NN) [64], support vector machine (SVM) [65], etc. Since SFS is used to find out the most influential feature dimensions for each phonetic transition category, we choose KNNR as the classifier in this study. This is because SFS procedure is usually time-consuming and KNNR is an efficient method as compared with other classifiers. Then, we performed a simple search to find the best value of K in KNNR is 9 in our experiment.

To evaluate the performance more objectively, we used leave-one-out (LOO) error [66] as the performance criterion, where a vector is selected as the test vector and all the other data as the training data. This process is repeated until each data point has served as the test vector. The final classification rate is the overall classification rate of these test vectors. KNNR with LOO is the most straightforward approach due to its simplicity, although other classifiers or performance criteria could also be used, too. After the use of SKL (SFS, KNNR, and LOO), we can obtain the training data set of each phonetic transition category and its corresponding classification rates. Since we have defined 54 kinds of phonetic transition categories, there are thus a total of 54 training data set to be used for boundary

refinement. In this study, the fundamental steps of SKL boundary refinement are listed as follows:

1) Collect several candidate boundaries located within [*N*, -*N*] ms with an interval of 2 ms around the initial boundary (obtained from HMM or HMM+DTW). In this study, *N* is assigned 50 for TTS-455 and 100 for SVS-1384, respectively.
2) For each candidate boundary, we extract its corresponding most influential feature dimensions which were identified via SKL feature selection in advance.
3) Finally, the boundary with the most amounts of "correct" votes is regarded as the optimal boundary, i.e., the refined boundary.

5.5. Performance Evaluation of Statistics-based Method

In our experiments, the speech corpus TTS-455 described in Chapter 4 was divided into two sets, including 300 sentences for the training set and 155 sentences for the test set. The boundaries of these sentences were labeled manually in advance. The training sentences were used to find out the most influential feature dimensions at the first and then were used to construct the KNNR classifier for later boundary refinement. The test sentences were then used to verify the performance. Fig. 5.4 demonstrates the experimental results. It is evident that the SKL boundary refinement definitely improves the performance of HMM-based alignment. (The closed test refers to the evaluation result of the training set, while the open test is the evaluation result of the test set.)

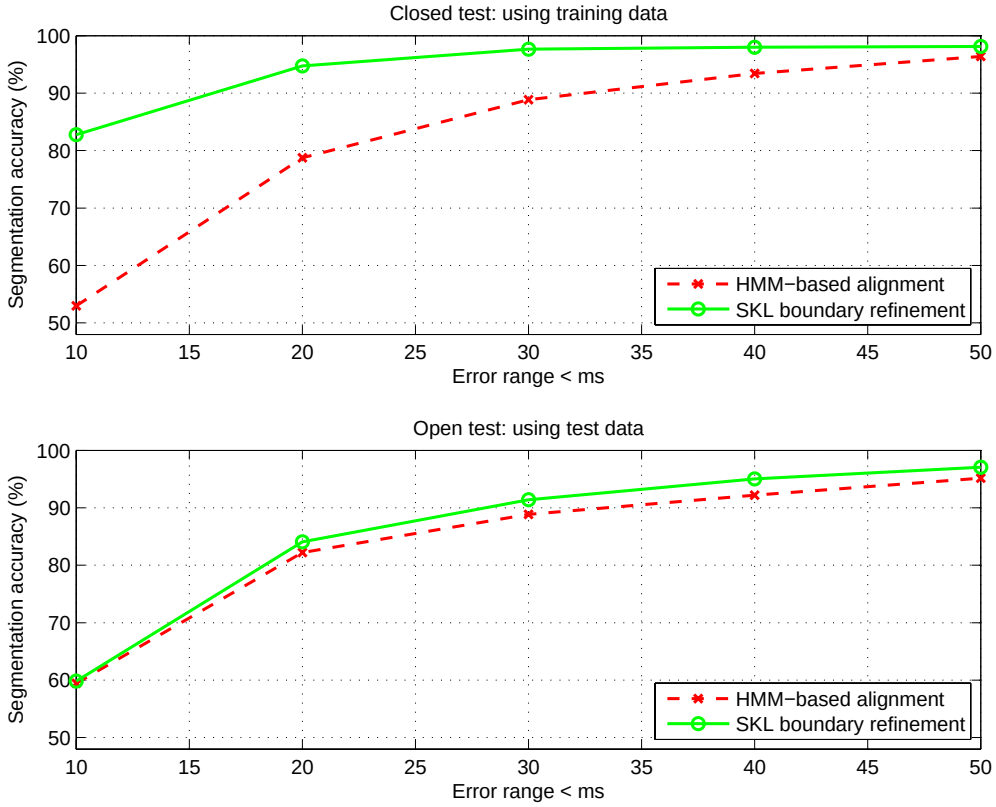


Fig. 5.4. Performance comparison between HMM-based alignment and SKL boundary refinement. Top: closed test. Bottom: open test. (Evaluted data: TTS-455)

On the other hand, we employ the same procedure mentioned above to evaluate the performance of the SKL boundary refinement on singing voice data. Similarly, the singing voice corpus SVS-1384 described in Chapter 4 was divided into two sets, including 800 sentences for the training set and 584 sentences for the test set. The boundaries of these sentences were labeled manually in advance. Fig. 5.5 shows the experimental results which indicate that the SKL boundary refinement can effectively refine the initial estimates obtained by HMM and DTW.

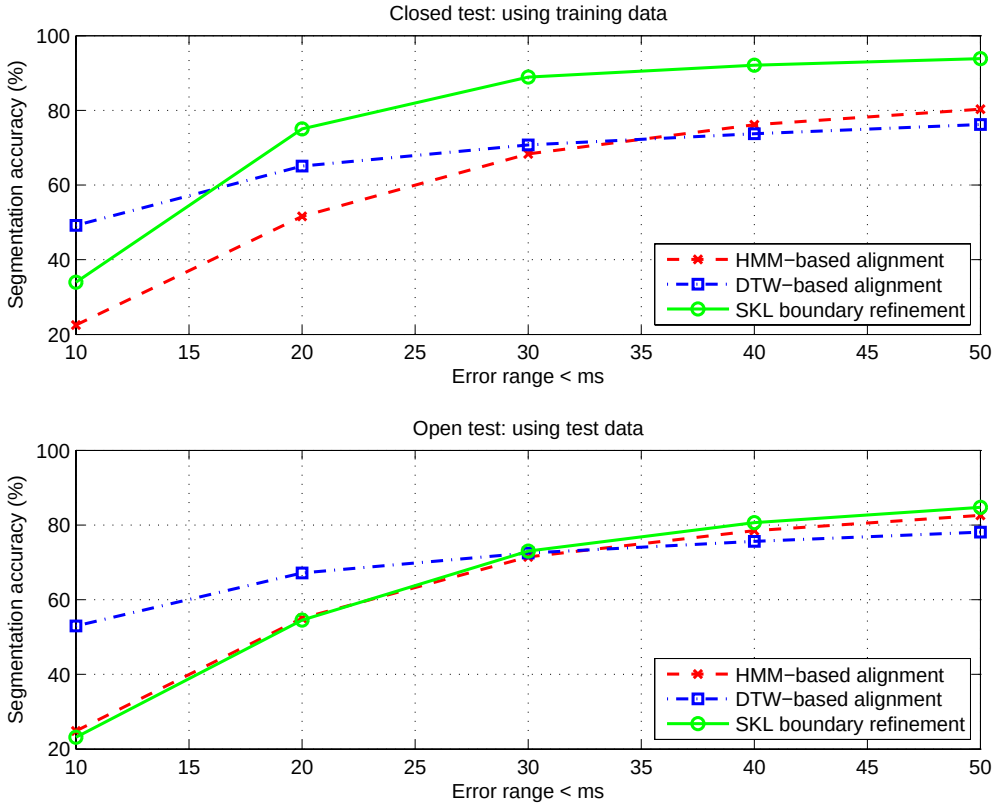


Fig. 5.5. Performance comparison among HMM-based alignment, DTW-based alignment and SKL boundary refinement. Top: closed test. Bottom: open test. (Evaluted data: SVS-1384)

5.6. Heuristic Method

As a matter of fact, not all phonetic transition categories show a satisfactory performance in our experiments. The segmentation accuracies of 20 ms tolerance of the first nine categories are worse than those of the rest. These categories correspond to “nine types of FINAL + first type of INITIAL”. Through visual and aural inspection we observed that the performance degradation of the first nine categories is mostly due to voiced-to-voiced co-articulation between syllables. However, refining the boundary of these categories is more complicated, and little related research has been reported in the literature. Here we propose a heuristic method for these categories. This method is based on

the observation that most boundaries labeled by humans are located in a region with lower log energy. In addition, the differences of formants between two sides of a true boundary are usually obvious. As a result, we utilize both log energy and formants to design a post-processing scheme in this study.

In theory, *F1* (first formant) and *F2* (second formant) can be calculated more precisely than *F3* (third formant) or other advanced formants, we thus take the two formants only to carry out the following procedures. Initially, we define two terms based on the ratio and the distance between *F1* and *F2*:

$$F_{Ratio} = \frac{F1}{F2}, \quad (5.7)$$

$$F_{Dist} = F2 - F1. \quad (5.8)$$

1) At the first, we collected all true boundaries for each phonetic transition category of the “FINAL + first type of INITIAL” group. (Here we expand FINAL to 38 kinds and the first type of INITIAL to 5 kinds, i.e., there are totally 38 x 5 (190) categories).
2) In each phonetic category, we calculated the F_{Ratio} and the F_{Dist} both left and right frames for these boundaries (The size of each frame was 20 ms). In other words, if there are N boundaries in certain phonetic category, then the formant-based data set of this phonetic category will be N x 4 dimensional matrix. Table 5.3 shows the formant-based data set:

TABLE 5.3 A FORMANT-BASED DATA SET.

	Left Frame		Right Frame	
	F_{Ratio}	F_{Dist}	F_{Ratio}	F_{Dist}
1st boundary	0.53	1613	0.09	2182
2nd boundary	0.51	1594	0.13	2365
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
Nth boundary	0.52	1620	0.12	2288

In the boundary refinement procedure, there are several steps described as below:

1) Calculate the F_{Ratio} and the F_{Dist} both left and right frames for a set of candidate boundaries and take these features to compare their corresponding formant-based data set for each phonetic category.
2) The formant difference between a test frame and a trained formant-based data set can be calculated via the following equations (5.9), (5.10), and (5.11).

$$F_{Ratio\_Difference} = \frac{\left| F_{Ratio}^{Test} - \hat{F}_{Ratio}^{Train} \right|}{\hat{F}_{Ratio}^{Train}}, \qquad (5.9)$$

$$F_{Dist\_Difference} = \frac{\left| F_{Dist}^{Test} - \hat{F}_{Dist}^{Train} \right|}{\hat{F}_{Dist}^{Train}}, \qquad (5.10)$$

$$FD = F_{Ratio\_Difference} + F_{Dist\_Difference}, \qquad (5.11)$$

where FD represents the formant's difference.

3) Since each candidate boundary has its two-sided left and right frames, the total difference could be calculated via (5.12).

$$TotalDifference = FD_{LeftFrame} + FD_{RightFrame}. \qquad (5.12)$$

4) Sort these candidates in ascending order by their total differences shown in (5.13) and keep top 3 candidates.
5) Use log energy to pick up the best one from top 3 candidates. Here the best candidate should have smallest log energy in theory.

5.7. Performance Evaluation of Heuristic Method

In order to validate the feasibility of the proposed heuristic method, we conducted another experiment, in which the heuristic boundary refinement was performed to refine those boundaries in the "FINAL + first type of INITIAL" categories. To verify the benefit of the proposed heuristic method, we compare the performance difference between the SKL boundary refinement and the heuristic boundary refinement further.

Experimental results demonstrated that using the heuristic boundary refinement has approximately 3% performance improvement as compared with using the SKL refinement on TTS-455 data in the open test (30 ms segmentation error tolerance). However, there is no obvious performance improvement while using the heuristic boundary refinement on SVS-1384 data. The phenomenon is probably caused by that the positions of formants will change (or shift) as the corresponding pitch varies greatly. For example, it can be seen from Fig. 5.6 that the distribution difference of the first formants between two syllables ("ㄚ") of different pitch is apparent.

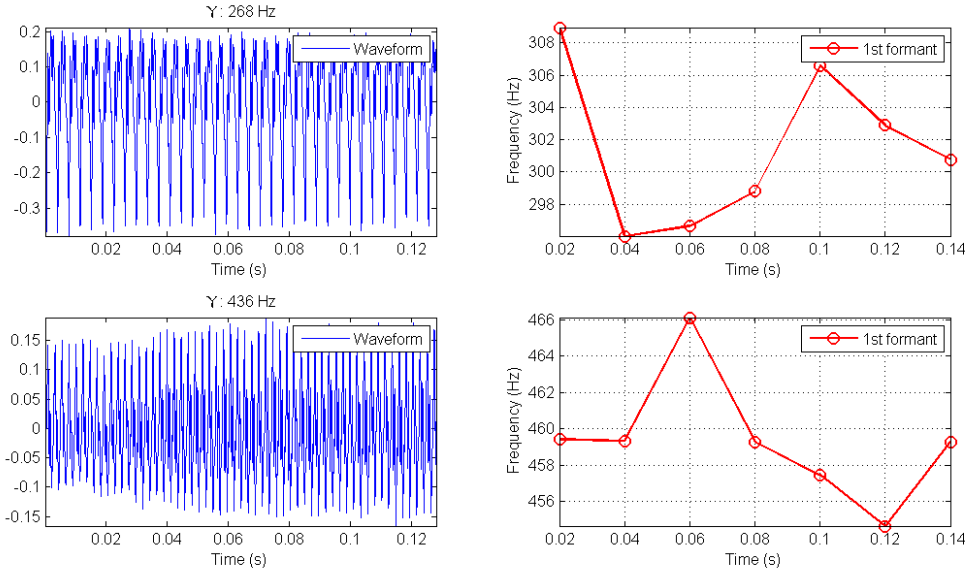


Fig. 5.6. Formants comparison between two singing voice data of different pitch (the same pronunciation, "ㄚ" ("a")).

Comparing two databases, TTS-455 and SVS-1384, we observed that there are usually larger pitch variations for the same syllables in SVS-1384. Therefore, it becomes doubtful whether taking the formants as the input features for the heuristic boundary refinement on SVS-1384 is reliable. In other words, it seems to lack generality to use such heuristic boundary refinement on different types of data. In the coming chapter, we will introduce a new concept based on a score prediction model to ameliorate the performance both on speech and singing voice data.