A Method of Multi-Layered Speech Segmentation Tailored for Speech Synthesis

Takashi Saito

Tokyo Research Laboratory, IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken, 242-8502, Japan
saito@jp.ibm.com

10.21437/Interspeech.2005-442

Abstract

This paper presents a speech segmentation scheme designed to be used in creating voice inventories for speech synthesis. Just the information about phoneme segments in a given speech corpus is not sufficient for speech synthesis, but multi-layers of segments such as breath groups, accent phrases, phonemes, and pitch-marks, are all necessary to reproduce the prosody and acoustics of a given speaker. The segmentation algorithm devised here has the capability of extracting the multi-layered segmental information in a distinctly organized fashion, and is fairly robust to speaker differences and speaking styles. The experimental evaluations with on speech corpora with a fairly large variation of speaking styles show that the speech segmentation algorithm is quite accurate and robust in extracting segments of both phonemes and accentual phrases.

1. Introduction

Creating well-formed voice inventories is, in general, time-consuming and laborious task. This has become a critical issue for speech synthesis systems that make an attempt to synthesize many high quality voice personalities or to customize specific voices. To make personalized voices easily realizable, automatic and robust techniques need to be devised for mass-producing voice inventories of various personalities. Therefore, quite a few automation methods for acoustic labeling (e.g., [1], [2]) and also for prosodic labeling ([3], [4]) have been investigated for speech synthesis.

In this paper, we propose a multi-layered speech segmentation method especially tailored for speech synthesis, in which the prosodic and acoustic features of a given target speaker are extracted simultaneously by making effective use of the text-to-speech synthesizer. Just the information about phoneme segments in a given speech corpus is not sufficient for speech synthesis, but multi-layers of segments such as breath groups, accent phrases, phonemes, and pitch-marks, are all necessary to reproduce the prosody and acoustics of a given speaker. The segmentation algorithm devised here has the capability of extracting the multi-layered segmental information in a distinctly organized fashion, and is fairly robust to speaker differences and speaking styles.

Another noteworthy feature of the presented segmentation method is that a new objective measure, *segmental reliability*, is introduced in the segmentation algorithm in order to optimize the results of phoneme segmentation by using phoneme validities of each segment, which are derived from the results of the preceding text analysis and pitch-mark detection stages. The new measure contributes greatly to the robustness of the segmentation method for wide variety of speaking styles as shown in the experimental evaluation.

Unlike the prevalent HMM-based approaches as in [1], [2], a dynamic time warping (DTW) based method is applied to the phonemic alignment in the core part of the segmentation method. One of the reasons why a DTW-based method is employed is that we attempt to establish a segmentation method that is well tailored to the target speech synthesizer by using the output of a speech synthesizer of the same family. Another reason is that we try not to use any kinds of training procedure, because it might become a barrier to a widespread use of the segmentation tool, for example, use by application software developers. From these points of view, we picked up a simple DTW-based phonemic alignment.

The following sections are organized as follows. In section 2, we describe the speech segmentation method in detail. Then, the experimental evaluation and discussions are provided in section 3. In the last section, we summarize the proposed method.

2. Multi-layered speech segmentation

2.1. Segmentation outline

Fig. 1 shows the process flow of the overall segmentation. Our previous waveform-concatenation-based TTS engine, ProTALKER® [5], is extended for use in this segmentation method to obtain a phonetic transcription and accentual phrases from input text, to predict phoneme durations for determining pause positions, and to generate a reference template for phonemic alignment.

First, text analysis for the given text is carried out using the text analysis module of the TTS engine to obtain a phonetic transcription and accentual phrases. The accentual phrase information is used in later stages for determining breath-groups and accentual phrases. Almost all of transcription errors are solved by just registering encountered unknown words in the TTS user dictionary.

Next, pitch-mark information is obtained through a wavelet-based pitch-marking method [6]. This is extracted, not just for use in the synthesizer, but also to provide voiced/voiceless information to the main segmentation stage that follows. The extracted pitch-marks are used to derive the segmental reliability, which is the key metric to optimize the process of the main segmentation stage.

The main segmentation stage is shown in Fig. 1 as the segmental optimization loop, in which the phoneme segmentation is carried out for each breath-group by optimizing in terms of segmental reliability. The detail of this procedure is described in the next section.

In the last stage of the segmentation procedure, the input utterances are segmented into accentual phrases by using the

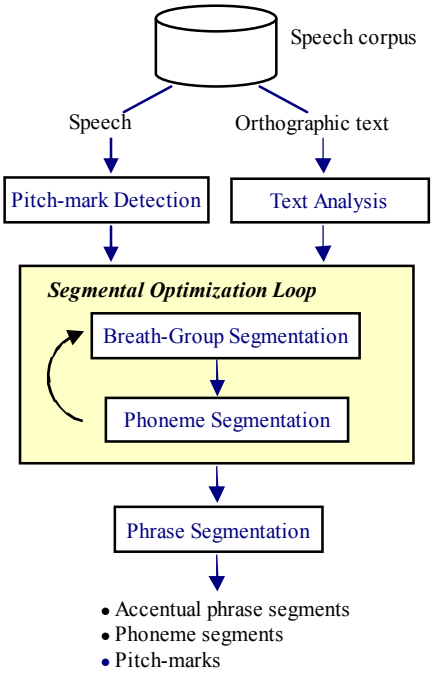

Fig. 1 Multi-layered speech segmentation tailored for speech synthesis

outputs of the text analysis stage and the phoneme segmentation stage.

2.2. Segmental optimization loop

This is the main segmentation stage, in which the breath-group and phoneme segmentation are carried out as an optimization in terms of the segmental reliability defined as follows.

2.2.1. Metric of segmental reliability

The segmental reliability Rs is defined as

$$Rs = 1 - \frac{\sum_{i=1}^{N} \delta_i}{N} \qquad (1)$$

where N denotes total number of segments in the input sentence and δ_i denotes the segmental validity of segment i, with the value is set to 0 if the segment contains no warnings about predefined check items, or otherwise set to 1. We defined 7 check items mainly related to the consistency between the phoneme assumed by the voiced/voiceless feature default and the actually obtained feature from the pitch-mark information. All the check items are listed as follows:

- If the phoneme is assumed from the transcription to be voiced, then it should have pitch-marks in the segmented region.
- If the phoneme is assumed from the transcription to be voiced, then it shouldn't have a pitch-mark starting point in the segmented region.
- If the phoneme is assumed from the transcription to be voiced, then it shouldn't have a pitch-mark ending point in the segmented region.
- If the phoneme is assumed from the transcription to be voiceless, then it shouldn't be filled with pitch-marks in the segmented region.
- If the phoneme transition is assumed from the transcription to be voiceless-to-voiced, then it should have a pitch-mark starting point in the transition region.
- If the phoneme transition is assumed from the transcription to be voiced-to-voiceless, then it should have a pitch-mark ending point in the transition region.
- If the segmented region is assumed from the transcription to be silence, then it shouldn't have a block of pitch-marks in the region.

2.2.2. Optimization procedure

The following steps describe the procedure for optimizing the phoneme segmentation. In the procedure, L_p is defined as a threshold value for the minimum length of intra-sentence pauses. In Step 2, silences detected in a sentence, which are longer than L_p, are picked up as intra-sentence pauses. A very rough setting is sufficient for the range of L_p. For instance, three values (110 ms, 190 ms, and 270 ms) determined in a preliminary experiment were actually used for L_p in the experiments described in later section. In Step 3, the pause positions in the phonemic transcription are automatically determined by an algorithm that uses the phoneme duration predicted by the synthesizer from the phonemic transcription and the silence positions obtained in Step 2. As a result, the input utterance is segmented into breath groups. This process plays an important role in the whole algorithm since it enables the next phonemic alignment of Step 4 to be simple and robust.

[Step 1] Set a threshold L_p, for the minimum length of intra-sentence pauses.
[Step 2] Find the intra-sentence pause positions by using the log power and spectrum with the threshold L_p.
[Step 3] Decompose the utterance into breath-groups by searching for the positions of accentual phrase boundaries that best match the intra-sentence pause positions obtained in the previous step. (A pause position determination algorithm is devised for this purpose.)
[Step 4] Divide each breath-group speech fragment into phonemic segments by doing phonemic alignment with the synthetic speech template. As the distance measure used in the phonemic alignment, the static and dynamic features of mel-cepstrum with normalized log power were used.
[Step 5] Validate all of the phonemic segments for the input sentence in terms of the segmentation reliability, and sum up the unreliable phonemic segments to calculate Rs.

The optimization scheme is to repeat from Step 1 to 5 for a predefined range for the threshold L_p, and take the best segmentation case as the final result, where the threshold gives the smallest number of unreliable segments, maximizing Rs.

3. Experimental evaluation

3.1. Speech materials

To evaluate the accuracy and robustness of the multi-layered speech segmentation ranging from the phonemic level to the phrasal level, we used a script set of 503 sentences from the ATR continuous speech database, and prepared 6 speech

corpora for the script uttered by professional speakers; S1F (female), S2F (female), S3M (male), N1F (female), C1F (female), and C2F (female). Each speaker was told to read the ATR script as follows:

- S1F, S2F, and S3M were asked to speak at a relatively slow speed.
- N1F was asked to speak at a normal speed.
- C1F was asked to speak in a character voice.
- C2F was also asked to speak in another character voice.

Table 1 shows the average utterance characteristics (speech rate, number of inserted pauses in a sentence, and F0 range) for the 6 speech corpora. The speech rate ranges from very slow speech (C2F, 6.15 [mora/sec]) to slightly fast speech (N1F, 8.35 [mora/sec]). In general, the speech rate and pause insertion rate are considered to have a negative correlation, and Table 1 supports this tendency in most cases. At the same time, it also shows a distinct exception to it: C1F (character voice), which is relatively fast speech, also has the biggest pause insertion rate. Just like the speech rate, the F0 range variation spans from normally flat (N1F, 1.10 [oct]) to fairly dynamic (C1F, 1.75 [oct]).

As stated above, even for the same script, a wide variation of utterance characteristics is actually observed in these 6 speech corpora. Here, we attempt to investigate the accuracy of the automatic speech segmentation and its robustness to speaker and style variations by using these corpora.

Table 1 Utterance characteristics in each speech corpus.

Corpus	Speech rate [mora/sec]	# of pauses per sentence	F0 range [oct] ([Hz])
S1F	6.82	2.16	1.27 (145 - 350)
S2F	6.67	2.12	1.18 (183 - 416)
S3M	7.16	2.27	1.55 (55 - 161)
N1F	8.35	1.99	1.10 (166 - 356)
C1F	7.39	2.43	1.75 (128 - 432)
C2F	6.15	2.39	1.33 (191 - 479)

3.2. Phoneme segmentation

We conducted the multi-layered speech segmentation for all the six speakers' data. In this experiment, we used three values (110 ms, 190 ms, and 270 ms) as the selectable values of L_p (threshold for minimum length of intra-sentence pauses) within its optimization range. L_p is optimized through the iteration loop to maximize the segmental reliability, R_s. In order to verify the effectiveness of the L_p optimization, we conducted the same segmentation procedure with a fixed value of L_p, whose value was selected for each speaker to give the best result for that speaker.

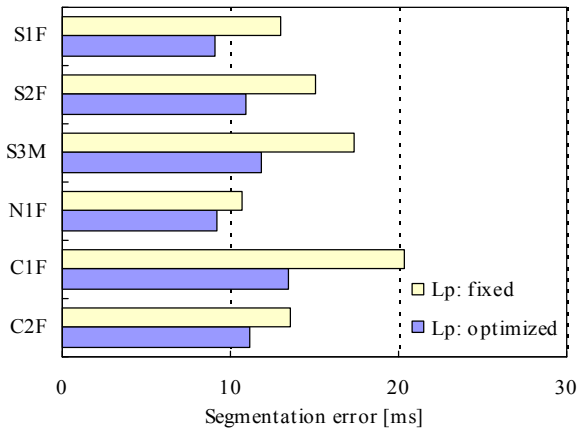

Fig. 2 Average phoneme segmentation error for each

We compared the phoneme boundaries in the six corpora (about 25,700 boundaries in each corpus) obtained by the automatic segmentation with those obtained by manual segmentation. Fig. 2 shows the averages of the phoneme segmentation errors, which means the differences between automatically segmented boundaries and manually segmented ones. "L_p: fixed" and "L_p: optimized" represent when L_p is fixed for each speaker's corpus, and when L_p is optimized through the proposed procedure, respectively. From Fig. 2, it can be seen that the average segmentation error for "L_p: optimized" is fairly small, and the differences among the speech corpora are also relatively small (from 9.1 to 13.5 ms). It can also be seen that L_p optimization contributed a lot to decrease the segmentation error for "L_p: fixed" (from 14.0% to 33.5%). This is especially prominent for corpora whose segmentation errors for "L_p: fixed" are relatively larger than the others, for instance, C1F or S3M. In view of the speaking styles, the corpora of the characters' voices (C1F and C2F) differ greatly from the others. Nonetheless, their segmentation results showed fairly good accuracy almost like the others.

As a whole, the phoneme segmentation worked successfully, in view of the intrinsic variations in manual segmentation. The robustness is very promising for applications with various speakers and speaking styles. Moreover, the stability of segmentation over speakers and speaking styles results in decreasing the number of unexpected error handlings in segmentation. Therefore, it also contributes to making the process of voice inventory creation simple and effective so that even non-specialists are able to easily work on it.

3.3. Accentual phrase segmentation

Accentual phrase segmentation is the final stage of the multi-layered speech segmentation as shown in Fig. 1. In this stage, the accentual phrase boundaries and accent types of each accentual phrase are automatically obtained by combining the results of the preceding two stages, phoneme segmentation and text analysis. Here, we investigated the accuracy of accentual phrase segmentation by comparing automatically obtained phrase boundaries with manually set ones for all of the 503 sentences of each corpus.

Accentual phrase segmentation errors were calculated as the coincidence rate between the automatically segmented phrase boundaries and the manually segmented ones at two levels: *the intermediate level* and *the final level*. At the intermediate level, the automatic and manual boundaries were checked to determine whether each has a boundary at the same phonemic label position. At the final level, the two kinds of boundaries that coincided at the intermediate level were checked to determine whether each has a boundary at the same time position within a certain time range (here we used a value of ±100 ms, which was the same value as used in [7]). The accuracy of the intermediate level is that of the text analysis stage, and the accuracy of the final level really means that of this final stage.

Table 2 shows the accentual phrase segmentation results. C_{int} , E_{insert}, and C_{fin} denote the percentage of correctly segmented accentual phrase boundaries at the intermediate level, the percentage of accentual phrase boundary insertion errors at the intermediate level, and the percentage of correctly segmented accentual phrase boundaries at the final level, respectively. These values are defined as

$$C_{\text{int}} = \frac{N_{c\_int}}{N_{bnd}} \text{ x } 100 \quad [\%] \qquad (2)$$

$$E_{insert} = \frac{N_{e\_int}}{N_{bnd}} \text{ x } 100 \quad [\%] \qquad (3)$$

$$C_{\text{fin}} = \frac{N_{c\_fin}}{N_{bnd}} \text{ x } 100 \quad [\%] \qquad (4)$$

where N_{bnd} is the total number of accentual phrase boundaries, N_{c_int} is the number of correctly segmented boundaries at the intermediate level, N_{e_int} is the number of incorrectly inserted boundaries at the intermediate level, and N_{c_fin} is the number of correctly segmented boundaries at the final level.

As seen in the table, over 90% of the boundaries of accentual phrases were correctly segmented at the intermediate level. This means that the text analysis module can produce very good default accentual phrase boundaries for phrase segmentation with a reasonably small insertion error rate, E_{insert}. On the other hand, the accuracies of accentual phrase segmentation at the final level decreased by 4% to 7% from those of the intermediate level. This is mainly because of intra-sentence pauses shorter than 110 ms, which is the minimum value of L_p (the threshold for the minimum length of intra-sentence pauses in the multi-layered segmentation). However, the decrease caused by short pauses does not affect the accuracy of the F0 unit extraction for accentual phrase segments, because only the F0 values of vowel positions in extracted phrase segments are used for F0 phrasal units.

Table 3 shows the accuracy of accent type setting for the correctly segmented accentual phrases at the intermediate level. The accuracy of accent type setting, C_{acc}, given by

$$C_{\text{acc}} = \frac{N_{acc\_cor}}{N_{phr\_cor}} \text{ x } 100 \quad [\%] \qquad (5)$$

where N_{acc_cor} is the number of correctly accent-type set phrases out of the correctly segmented accentual phrases, and N_{phr_cor} is the number of correctly segmented accentual phrases.

Since the accent type setting is done by the text analysis module, the accuracy comes mostly from the performance of the text analysis and partly from mismatches that occurred between the script and the actual utterance. Although the accent type labeling is purely on a top-down basis, the output is good enough to be used as an initial labeling value.

Table 2 Accuracy of accentual phrase segmentation for each corpus.

Corpus	C_{int} [%]	E_{insert} [%]	C_{fin} [%]
S1F	93.5	3.2	86.7
S2F	95.2	6.5	88.8
S3M	94.7	5.1	88.5
N1F	93.3	4.4	89.3
C1F	96.5	8.1	92.1
C2F	93.1	4.5	89.2

Table 3 Accuracy of accent type setting for each corpus.

Corpus	S1F	S2F	S3M	N1F	C1F	C2F
C_{acc} [%]	93.1	90.5	87.9	86.4	92.1	91.6

4. Concluding remarks

In this paper, we presented an effective and robust multi-layered speech segmentation method tailored for speech synthesis. Through experimental evaluations on 6 speech corpora, we showed that the proposed speech segmentation method has a good performance in segmenting and labeling from phoneme to phrase level. Moreover, we showed that it is robust against variations of speakers and speaking styles. Since the proposed method needs essentially no training on speech data, it can be used for a new corpus without any additional cost. These promising results support the idea that the segmentation method might contribute largely to reduce the time and cost to create a new voice inventory. Actually from the preliminary results in three cases of feasibility studies on this method, we found that the duration of voice inventory creation for a speech database of about three thousand sentences was reduced to about one month by using this method, while it took at least three months in the conventional procedure. Since the proposed multi-layered speech segmentation method can be used for a generic speech database construction, it is expected to be quite helpful not only for speech synthesis, but also general research related with speech corpora.

5. References

[1] R. Donovan and E. Eide, "The IBM trainable speech synthesis system," *Proc. ICSLP*, pp. 1703-1706, 1998.

[2] Yeon-Jun Kim, Alistair Conkie, "Automatic segmentation combining an HMM-based approach and spectral boundary correction," *Proc. ICSLP*, 2002.

[3] A. Sakurai et al., "A linguistic and prosodic database for data-driven Japanese TTS synthesis," *Proc. ICSLP*, pp. 1048-1051, 1998.

[4] A. Syrdal and J. Hirschberg, "Automatic ToBI prediction and alignment to speed manual labeling of prosody," *Speech Communication*, vol. 33, pp.135-151, 2001.

[5] T. Saito, M. Sakamoto, Y. Hashimoto, M. Kobayashi, N. Nishimura, and K. Suzuki, "ProTALKER: a Japanese text-to-speech system for personal computers," IBM TRL Research Report, RT0110, June 1995.

[6] M. Sakamoto and T. Saito, "An automatic pitch-marking method using Wavelet transform," *Proc. ICSLP*, pp.650-653, 2000.

[7] K. Iwano and K. Hirose, "A statistical modeling of fundamental frequency contours in moraic unit and its use for the detection of prosodic word boundaries," *IPSJ Trans.* Vol. 40, No. 4, pp.1356-1364, Apr. 1999. (In Japanese)