Abstract—In this paper, a fast algorithm for text-dependent speaker recognition is proposed for real-time applications on embedded platforms. Considering the similarity and effectiveness, we introduce the Non-Linear Partition (NLP) method used for isolated word speech recognition to text-dependent speaker recognition, where both training and verification can be performed in terms of segments instead of frames and which can save the execution time compared with dynamic programming based methods. However, due to the segmentation rule of distance accumulation, NLP is not robust enough, often resulting in an unreasonable segmentation result in case of noise. Accordingly, the simple rigid application of NLP will cause an equal error rate (EER) increase by 23.6%. To address this problem, an improved NLP algorithm is proposed, in which an utterance is segmented according to the Mahalanobis distance measure of between-frame changes. Experimental results show that the proposed algorithm can significantly improve the text-dependent speaker recognition in both recognition performance and computational efficiency.

I. INTRODUCTION

As one of the most popular speaker recognition technologies, text-dependent speaker recognition has the feature of dependable performance and simple implementation. If it can be ported to embedded devices, for example, a Pocket PC (PPC), it is obvious that this technique can be more widely used. The difficulty here, however, is that embedded devices are usually weak in available resources, such as the processor speed and the memory size. Meanwhile for reducing the chip size and power dissipation, floating-point arithmetic units are usually not provided [1]. As a typical PPC, dopod P800 has a CPU with the frequency of only 201MHz, with only one fixed-point arithmetic unit. For speaker recognition involves an enormous amount of floating-point operations, if a recognition system over PC is directly ported to PPC, the execution time for training and verification will be too much. On dopod P800, for instance, it takes about 10 seconds to train a template for a three-Chinese-character utterance, and as long as five seconds to verify. (Details of the experiment are provided in section 4.) Obviously, it is extremely unacceptable for real-time applications.

There is something in common after considering the methods used in text-dependent speaker recognition. In most prevalent methods, like template matching and Hidden Markov Models (HMM), each feature vector is computed meticulously. For template matching, the model consists of a template which is a sequence of vectors extracted from the utterance. During verification, dynamic time warping (DTW) is often used to calculate the distance between each test phrase feature and a speaker template [2]. DTW [3, 4] is a typical algorithm that is used to match the test utterance with the trained template. For the HMM method, HMMs that encode the temporal evolution and model statistical variation of the features are used to train the speaker templates. During verification, the likelihood of each test feature is calculated against the speaker’s HMMs [2]. It can be seen that every feature vector is meticulously used for both training and verification. This is reasonable for making an efficient use of information extracted from utterances. However, although the computation is acceptable for a desktop PC, it is beyond the capability of most common embedded devices.

To port speaker recognition systems into embedded devices, some code optimization, such as the floating-point code conversion, was proposed to reduce the execution time in [5]. There are also some implementations of speech recognition over fixed-point digital signal processors [6, 7]. However, the research on speeding up text-dependent speaker recognition is less. In the view of the relationship between speech recognition and speaker recognition, there may exist a certain method to reduce the computational complexity. Indeed, many of the same methods successfully applied to speech recognition have also been used for speaker recognition [2]. In this paper, the Non-Linear Partition algorithm used for isolated word speech recognition is referenced.

II. NLP AND THE ALGORITHM PROPOSED

A. NLP

In the area of speech recognition, Non-Linear Partition (NLP) is a fast pre-selection algorithm for isolated word speech recognition especially for the embedded applications by decreasing the amount of information to be processed [8]. According to the change of voice information, the algorithm divides an utterance into several segments. Frames in the same segment are expected to be similar. Therefore, training and verification can be implemented by segment other than by frame, which compresses the information and saves the execution time.

Assuming X = (x1, …, xt) represents the feature sequence extracted from an input utterance, where xt (1 ≤ t ≤ T) is a k-
dimensional feature vector for frame $t$. The distance between $x_t$ and $x_{t+1}$ is defined as

$$d_t = d_{cep}(x_t, x_{t+1}) = \sum_{k=1}^K [W_k(x_t - x_{t+1})]^2$$

$$1 \leq t \leq T - 1,$$  \hspace{1cm} (1)

where $w_k$ is the weight of dimension $k$.

First, the average of the overall changes is calculated, assuming that each utterance is divided into $N$ segments:

$$\Delta D = \frac{1}{N} \sum_{i=1}^{T-1} d_i$$  \hspace{1cm} (2)

Second, if $n_i (1 \leq i \leq N, n_0 = 0)$ meets the following condition:

$$\sum_{i=1}^n d_i \leq i \Delta D \leq \sum_{i=1}^n d_i,$$  \hspace{1cm} (3)

then the segmentation rule of NLP indicates that the original feature vectors from $x_{n_{i-1}} + 1$ to $x_{n_i}$ form a new segment $i$ [9].

### B. The Improved Algorithm

However, a flaw of the segmentation rule was found during the experiment. It is not robust to cases where there exists noise.

Fig. 1 a) gives two original utterances of the same content spoken by the same speaker. It can be seen that they are almost the same except that utterance (b) contains some noise in the beginning (although the utterances were recorded in clean condition, such tiny noise was acceptable). Generally, similar segmentation results are expected for these utterances, but it turns out not. In the case of $N = 4$, results are as bellow:

- Utterance (a): \{n_1 = 33, n_2 = 22, n_3 = 34, n_4 = 20\},
- Utterance (b): \{n_1 = 14, n_2 = 37, n_3 = 15, n_4 = 43\},

where $n_i (1 \leq i \leq 4)$ indicates the amount of frames in segment $i$, which gives three demarcation points of the input voice respectively at frame $n_1$, $n_1 + n_2$, and $n_1 + n_2 + n_3$. It is noticed that the total effective frames for both utterances are 109 ($n_1 + n_2 + n_3 + n_4$), but the frame distributions into segments for these two utterances are quite different, resulting in an unsuccessful verification.

The difference of frame distributions is mainly due to the noise. To verify this speculation, the following experiment was made: first, the same noise as in utterance (b) was inserted into the beginning of utterance (a); second, the new assembled utterance (a') was segmented by NLP. It can be found that the frame distributions of utterance (a') is similar as those of utterance (b). The segmentation results are shown in Fig. 1 b).

The reason causing the above phenomena is that the method of distance accumulation used for the segmentation rule can be easily influenced by noise. In this case, a little extra noise will cause a big distance between the noise frames and the effective speech frames, which makes the segment containing such frames has a rapid distance accumulation and results in a great change for the frame distributions into segments.

To address this problem, an improved NLP algorithm is proposed, where the segmentation rule is different from the original one. Instead of summing up the changes of voice information, a new rule is proposed to find the demarcation points of voices in a more reasonable way. Since it is assumed that frames in the same segment are similar in NLP, it can be considered that there must be a great difference between adjacent segments, and demarcation points may exist at the frames where segments change. Then if demarcation points have been found out, the utterance can be divided into segments according them. As a result, frames between two adjacent demarcation points are grouped to be a segment, without considering how many changes accumulated for these frames.
In order to find demarcation points, a demarcation score is proposed for each frame. For frame $i$, the demarcation score $s_i$ is defined as:

$$
s_i = \begin{cases} \frac{(s_i - \bar{s})^2}{\sigma}, & \text{if } (s_i - \bar{s}) > 0 \\ 0, & \text{else} \end{cases}$$

where $d_{ij}$ is the distance between features $i$ and $j$, $M$ is an empirical constant equaling to the width of the window, $\bar{s}$ and $\sigma$ are the mean and variance of sequence $s$, respectively.

Generally speaking, the larger the demarcation score is, the more probable the frame is a demarcation point. Fig. 2 shows the demarcation scores of the same utterances showed in Fig. 1 a). We can see in the illustration that the demarcation points are apparent and the segmentation will be more reasonable.

III. SYSTEM IMPLEMENTATION

A text-dependent speaker recognition system is normally composed of a training procedure and a verification procedure, which can be separately demonstrated as below.

A. Training

For each training utterance, the training procedure is as follows:

1. Extracting feature vectors from the input utterance;
2. Dividing the feature vectors into $N$ segments ($S_1, S_2, \ldots, S_n$) using the NLP algorithm;
3. Training a template $M_i$ ($1 \leq i \leq n$) for each segment;
4. Combining the templates into a training model.

B. Verification

For each test utterance, the verification procedure is as follows:

1. Extracting feature vectors from the input utterance;
2. Dividing the feature vectors into $N$ segments ($S_1, S_2, \ldots, S_n$) using the NLP algorithm;
3. For segment $i$ ($1 \leq i \leq N$), calculating the verification score against template $i$ for all frames in this segment;
4. Carrying out a symmetrical test if the first judgment is accepted.

Fig. 3 shows the implementation of the system.

IV. EXPERIMENTS

A. The Database

A database of 96 speakers was prepared for the experiment, which included balanced male and female speakers. For each speaker, 10 different Chinese words were uttered for 5 times. Every word contained only 2 or 3 Chinese characters. The feature vectors used in the system were 16-dimensional MFCCs and its first-order derivative. The sample rate was 16 kHz. All utterances were recorded in clean condition.

During a specific experimental procedure, training utterances were individually divided into two segments in a uniform way. The frame with the maximal demarcation score was selected as the demarcation point (For an utterance starting and ending with tiny noise, any frame in the first $k$ and the last $k$ ones was not considered as a demarcation point, where $k$ was assigned as one fifth of the length of the frame sequence). Each segment was used to train a Single Gaussian Model (SGM), and the two SGMs were combined into the training model. In the verification procedure, every test utterance was divided in the same way with a likelihood score calculated for each segment. At last, an empirical threshold was introduced to decide whether or not to carry out a symmetrical test.

For every speaker, 2 utterances were randomly chosen from 5 for each word and used for training, each utterance for a model. The rest were used for verification. This created 5,760 target trials and more than 5,000,000 imposter trials.

---

Fig. 2: Comparison chart for demarcation scores of the two pieces of utterances showed in Fig. 1 a), where demarcation points are apparent, while the influence raised by noise disappears almost completely.

Fig. 3: The flow chart of the text-dependent speaker recognition system, where operations before feature extraction are omitted.
B. Equal Error Rate (EER)

Fig. 4 illustrates the performance of text-dependent speaker recognition for different algorithms where the DTW-based system is the baseline. Here the DANLP-system indicates the system using the traditional distance accumulation NLP, while the MDNLP-system for proposed Mahalanobis distance NLP algorithm.

As can be seen from Fig. 4, a simple rigid transplant of NLP led to a degradation of performance by 23.6% in EER increasing, with EER rising from 2.08% to 2.57%. The use of improved NLP algorithm improved the performance. The MDNLP-system made an EER decrease by 17.3% from 2.08% to 1.72% compared with the baseline system. Meanwhile, it outperformed the DANLP-system by 33.1%.

The calculation is performed frame by frame in a DTW system, while segment by segment in an NLP system. Generally speaking, if the segmentation is reasonable, the segment-based training and verification will be more robust than the frame-based. This can explain why a DANLP-system outperformed a DTW-system.

C. Time Cost

Table 1 displays the time cost of the DTW-based system and that of the MDNLP-system on dopod P800. During the test, 15 pieces of utterances were chosen randomly from the database. The two systems were used separately for training and verification, and the individual time overhead was recorded afterwards. It can be concluded from the table that the MDNLP-system’s time cost was much less than that of the baseline DTW-based system both in training and verification.

What is especially worth to point out is that, when adopting multi-pass training, the verification time cost will rise linearly due to the linear increase of templates preserved by the baseline system caused by the increase of training times. However, for the MDNLP-system, multi-pass training will not lead to template increase. As a result, the verification time cost won’t grow with the increase of training times. It’s another advantage of the new system in real-time applications.

V. Conclusions

In this paper, a fast algorithm is presented for text-dependent speaker recognition systems by referencing and improving the NLP algorithm from isolated word speech recognition. A new segmentation rule is used in the proposed algorithm, which makes the segmentation more reasonable and results in a much better verification rate. At the same time the segmentation modeling method saves a lot of execution time. In a word, the MDNLP-system is faster and more accurate than the baseline DTW-system, making the text-dependent speaker recognition system more responsive to the embedded platform applications.

REFERENCES


![Fig. 4: The Detection Error Trade-offs (DET) curves for the text-dependent speaker recognition system.](image-url)