ON INTEGRATING TONAL INFORMATION INTO CHINESE SPEECH RECOGNITION

Xia WANG, Yuan DONG
Nokia (China) Research & Development Center, Beijing
Email: {xia.s.wang, yuan.dong}@nokia.com

Juha Iso-Sipilä, Olli Viikki
Nokia Research Center, Speech and Audio Systems Laboratory, Tampere
Email: {juha.iso-sipila, olli.viikki}@nokia.com

ABSTRACT

Reliable pitch detection is important in Chinese speech recognition since Chinese is a tonal language. In this paper, several pitch information integration approaches are investigated. In a noise-free environment, conventional pitch estimators work quite well. In adverse conditions, however, robustness of pitch detection algorithms is still a challenging problem. Our experimental results show that by using pitch information, a performance improvement can be obtained in a clean environment. However, a substantial recognition accuracy degradation is observed in adverse conditions due to the noise sensitivity of pitch estimators. Our experimental results indicate that front-end extracting higher-order cepstral coefficients provides the best results when testing the recognition performance in Chinese.

1. INTRODUCTION

While Pitch Determination Algorithms (PDAs) have widely been used in speech coding, their utilization in speech recognition has been fairly modest. Pitch has been an important feature in coding and prosody modeling for decades [1][2][11]. Recently, more interest has been focused on the usage of pitch information in Automatic Speech Recognition (ASR) [5][9]. Since Chinese is a tonal language, tone integration into an ASR system has nevertheless been researched over years.

In Mandarin Chinese, there are five lexicon tones, each of which has a distinct pitch contour. In Chinese ASR, the use of pitch and tone information generally falls into two categories, namely front-end and back-end integration. In front-end integration, pitch information is treated as a part of feature vector coefficients [8][9]. Pitch, voicing features and their time-derivatives are normally used in such approaches. In back-end integration, separated tone and syllable recognition was commonly used at beginning, in which base syllable recognition is based on traditional HMM models, and tone recognition is done separately by statistical discriminative methods, neural networks, or even HMMs. A decision is given as the final recognition result based on the combination of the two channels. This kind of approaches cannot take tone into consideration in the Viterbi decoding procedure of syllable recognition; therefore they are neither efficient nor natural. More integrated approaches such as tonemes in [8] or decision tree approach in [10] were proposed later. All the above approaches have been reported to produce performance improvements. However, it is worth mentioning that no results in adverse conditions have been reported in the literature.

In this study, we used a speaker-dependent, small vocabulary, name dialling system. The most distinguishing feature is its noise robustness requirement due to its challenging operating environment where the Signal-to-Noise Ratio (SNR) can vary from 30 to –10 dB. Generally, voice dialling cannot be assumed to only be done in a noise-free condition. In practice, pitch estimation in noise is still a challenging problem, although in clean, many algorithms are reported to provide a quite good performance [2]. In section 2, our efforts on this issue are described.

Higher-order cepstral coefficients (HOC) have previously been reported to provide a good representation of pitch in a Chinese name dialling task [4]. The use of HOCs is compared to other tone representation techniques in Section 3.

2. ALGORITHM DESCRIPTION

In our research, a short-term autocorrelation-based PDA is implemented as our basic pitch estimator, which is found to work fairly well for clean utterances. However, when noise is present, its performance starts to degrade rapidly. To improve its noise-robustness, dynamic programming (DP) [2] is utilized as a post-processing approach. DP is widely used in pitch tracking and reported to be robust against noise [2][11]. It also works well in our experiments, providing good pitch estimates in moderate noisy conditions. But it also fails to provide an accurate estimation when SNR falls below 0 dB.

Without a reliable pitch estimator, tone classification loses its foundation. Therefore, direct tone integration into back-end is impractical for adverse speech recognition. As for front-end integration, voicing features are commonly used as complementary features of spectral parameters [5][9]. Although voicing features, Periodicity and Jitter (P&J), appended to MFCCs (Mel-Frequency Cepstral Coefficients) as a part of feature vector coefficients slightly improve the recognition accuracy in noise-free conditions, an obvious performance degradation is observed in adverse conditions in our experiments.

In this section, we will describe our pitch detection, pitch tracking and voicing feature extraction algorithms in detail.

2.1 Pitch Determination Algorithm

The basic pitch estimator used in our experiments is a short-term PDA used in MELP speech coder [1], based on the normalized autocorrelation function \( r(\tau) \), which is defined as follows,

\[
r(\tau) = \frac{c_{\tau}(0,\tau)}{\sqrt{c_{\tau}(0,0)c_{\tau}(\tau,\tau)}}
\]

(1)
where
\[
c_{\tau}(i, j) = -\left[\frac{\tau}{2}\right] \log \left[ \frac{M}{2} \right]_{i} - \sum_{k=\left[-\frac{\tau}{2}\right]}^{\left[\frac{M}{2}\right]_{j}} S_{k+i} S_{k+j}
\]

\( S_{j} \) is the \( i \)th sample of the low-pass filtered input signal.

The initial pitch estimate (fundamental period \( T_{0} \)) is defined as the lag value \( \tau \) that maximizes \( r(\tau) \), \( \tau = K, ..., M \), i.e.,
\[
T_{0} = \arg \max_{\tau} r(\tau)
\]

(3)

where \( K \) is the minimum lag, \( M \) is the maximum lag. Generally pitch frequency in normal voiced speech is 40-500Hz. Since \( f_{s} = \frac{f_{s}}{\tau} \), where \( f_{s} \) is the sampling frequency, i.e. 8000 kHz in our experiments, \( \tau \) can vary from \( K=20 \) to \( M=160 \).

Pitch value estimated by Equation (3) is further refined by band-pass voicing analysis, which filters the input speech signal into five frequency bands, i.e. 0-500, 500-1000, 1000-2000, 2000-3000, and 3000-4000. A better pitch value is searched near the two pitch estimates of current and previous frame, using the 0-500 filter output signal. The frame voicing decision is also made in this module based on the band-pass voicing status.

At last, low-pass filtered LPC residual signals are analyzed to search for the final pitch value near the refined pitch candidate in band-pass voicing analysis.

### 2.2 Pitch Tracking

The PDA described above has sometimes doubling or halving errors, although an pitch doubling check procedure has already been embedded. If the estimated pitch value is multiples of the actual pitch, we call it a doubling error. On the other hand, if the pitch estimate is half of the actual pitch, it is a halving error. Most periodicity detectors suffer from doubling and halving errors. If context and known language knowledge are taken into consideration, these errors can be corrected. The greatest problem in pitch tracking is its sensitivity to ambient background noise. When SNR falls below 0 dB, it starts to give almost all zero pitch estimates. To improve its noise robustness, dynamic programming is utilized in post-processing [2][3][12]. First, more than one pitch candidates, say \( j \), are selected, which produce the first \( j \) peaks of \( r(\tau) \). Dynamic programming is then performed to try all possible pitch tracks and select the best one with the minimal cost. Although DP introduces some delay in the system, it substantially improves the accuracy of pitch detection.

Suppose at each time instant or frame index \( i \), there are \( S_{i} \) states numbered from 0 to \( S_{i}-1 \), either unvoiced (state 0) or voiced (state 1 ... \( S_{i}-1 \)). The voiced states correspond to the initial pitch candidates. The block diagram of applying dynamic programming to pitch tracking is shown in Figure 1 [12].

For each state \( j \) at time instant \( i \), two penalty functions need to be defined, i.e. a local cost value \( d_{ij} \) and a transition cost \( \delta_{i,j,k} \) indicating the cost when transitioning from previous time instant's state \( k \) to current state \( j \). The cumulative cost \( D_{ij} \) is calculated according to equation (4) [3],
\[
D_{ij} = d_{ij} + \min\{ D_{i-1,j} + \delta_{i,j,k} \}, 1 \leq j \leq S_{i}
\]

(4)

For all the possible state transitions between previous and current time instant, only the one that produces the lowest accumulative cost is saved as a back pointer. After a certain delay, backtracking can be done by following the back pointers on the lowest cumulative cost path.

Although the improved pitch tracking algorithm performs well in both clean and moderately noisy conditions, it also fails to detect the periodicity of input speech signals under very noisy conditions.

### Figure 1. Dynamic programming applied to pitch tracking

#### 2.3 Voicing Feature Extraction

It has been reported that voicing features, Periodicity and Jitter (P&J), and spectral information are complementary and that improved speech recognition rate is obtained by combining the two sources of information [5]. This idea is widely adopted and applied in [9].

Both periodicity and jitter are derived from pitch analysis. Periodicity is the largest peak of normalized autocorrelation function \( r(\tau) \), \( \tau = K, ..., M \), while jitter is the variation in estimated pitch values between consecutive frames. For voiced speech, periodicity is about 1.0 and jitter is close to zero.

In our experiments, P&J were appended to feature vectors. Their first-order time derivatives were also calculated in the same way as for cepstral coefficients.

### 3. DATABASE AND EXPERIMENTS

#### 3.1 Database

The Chinese in-house name database "Pitch99" was used in all the experiments, which is designed for verification of pitch estimator and tone classifier. It consists of 128 pairs of
easily confusable Chinese names, in which 3-syllabic names (long names) and 2-syllabic names (short names) are in half. Ten male and ten female speakers took part in recordings in a quiet office environment. The two names within a pair are identical if the tone is ignored. Here are two examples, “fu4 zuo4 yi4” vs. “fu2 zuo4 yi4”, and “bao1 yun2” vs. “bao4 yun2”. To simulate the typical usage pattern of a name dialling system, different recording sessions of one speaker were arranged on different days.

3.2 Recognizer

In the baseline system, 12 MFCCs, log-energy, and their first-order time derivatives (delta cepstral coefficients) were extracted as feature vectors. A recursive feature vector normalization approach was utilized in front-end to provide noise-robust parameters [6]. A left-to-right HMM model with state-duration constrains [7] were trained using a noise-free utterance for each name-tag. Car noise was mixed into the clean utterances at a certain SNR to produce noisy test data. The average recognition rate of the baseline system was about 94% in clean, 78% in –5 dB across all speakers when tested on the Pitch99 database. Over 85% recognition errors were in-pair errors, i.e. tone errors.

3.3 Experimental Results

3.3.1 Pitch estimation

Although our basic PDA has a good performance in clean environments, it has problems on voicing detection in adverse environments. In figures 2 and 3, pitch value of unvoiced speech, including silence, is set to zero for a clear visual presentation. With DP, voicing detection is changed from a sub-band energy based solution to a by-product of dynamic programming in pitch tracking. Therefore not only the robustness against noise is improved, but also the accuracy of voicing detection.

The following figures show the pitch (T0) contour of ‘bao1yun2’ of a female speech, in clean and noisy (0dB) condition respectively. Figure 3 reveals that the basic PDA does not provide reasonable results at 0 dB. After DP tracking, although the pitch contour of ‘bao1’ is correctly estimated, that of ‘yun2’ is totally lost. All frames with a zero pitch value are labeled as ‘unvoiced’. Tone classification is unlikely to be noise-robust if it is established on such an unreliable pitch estimator. Our experiments also show that treating pitch as one dimension of feature vector only makes the recognition results degrade in adverse conditions. Direct pitch or tone integration is proven to be impractical due to our task specialty.

As we analyze the autocorrelation function of speech and noise, we notice that correlation function of noise degrades quite slowly. Sometimes it even looks periodic, which contravenes the basis of detecting periodic signal from noise, i.e., the correlation of noise degrades rapidly, while that of signal is periodic.

Periodicity and jitter have received much awareness recently since they were proposed in 1998 [5] and have been applied in different application areas. We also tried to utilize voicing features in our name dialling system as a means of pitch information integration. In Tables 1 and 2, we can see that recognition rate does increase in noise-free conditions. Unfortunately in adverse conditions, no improvement is observed. One reason is the noise-sentivity of voicing features, see Figures 4 and 5. Another lies in the mismatch of training and testing data, i.e. voicing features estimated in clean utterance could not model the noise-corrupted case accurately enough.

4. CONCLUSION AND OUTLOOK

The integration of tonal information is essential in Chinese speech recognition. Our investigation in several pitch information integration approaches reveals that robustness of pitch estimation algorithms is the major bottleneck in their use. The used pitch extraction algorithms are designed for speech coding initially. Due to the application difference of
coding and recognition, it might be necessary to search for more robust PDAs suitable for recognition tasks. On the other hand, higher-order cepstral coefficients have been found to provide an improved performance in Chinese and they can thus be considered a viable method for characterizing tonal information. However, the recognition rate under very noisy environments has still much room for improvements. In the future, we will focus on the combination of different techniques and knowledge sources instead of pure pitch analysis to enhance the noise robustness of tone information.

Table 1. Short name recognition rate comparison between the baseline, pitch, voicing features integration into front-end, and HOC front-end.

<table>
<thead>
<tr>
<th>Env.</th>
<th>Baseline (%)</th>
<th>+ Pitch (%)</th>
<th>+ P&amp;J (%)</th>
<th>+ HOC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>94.16</td>
<td>94.03</td>
<td>94.38</td>
<td>96.24</td>
</tr>
<tr>
<td>5 dB</td>
<td>90.60</td>
<td>89.96</td>
<td>90.54</td>
<td>93.39</td>
</tr>
<tr>
<td>0 dB</td>
<td>87.12</td>
<td>85.19</td>
<td>85.42</td>
<td>90.49</td>
</tr>
<tr>
<td>-5 dB</td>
<td>77.96</td>
<td>75.78</td>
<td>75.85</td>
<td>82.21</td>
</tr>
<tr>
<td>Average</td>
<td>87.46</td>
<td>86.24</td>
<td>86.55</td>
<td>90.58</td>
</tr>
</tbody>
</table>

Table 2. Long name recognition rate comparison between the baseline, pitch, voicing features integration into front-end, and HOC front-end.

<table>
<thead>
<tr>
<th>Env.</th>
<th>Baseline (%)</th>
<th>+ Pitch (%)</th>
<th>+ P&amp;J (%)</th>
<th>+ HOC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>94.19</td>
<td>94.08</td>
<td>94.44</td>
<td>96.14</td>
</tr>
<tr>
<td>5 dB</td>
<td>91.01</td>
<td>89.80</td>
<td>89.53</td>
<td>93.12</td>
</tr>
<tr>
<td>0 dB</td>
<td>86.95</td>
<td>85.52</td>
<td>85.91</td>
<td>91.26</td>
</tr>
<tr>
<td>-5 dB</td>
<td>78.07</td>
<td>78.02</td>
<td>76.97</td>
<td>84.87</td>
</tr>
<tr>
<td>Average</td>
<td>87.56</td>
<td>86.86</td>
<td>86.71</td>
<td>91.34</td>
</tr>
</tbody>
</table>

REFERENCES


