Auto-segmentation based VAD for robust ASR

Yu SHI, Frank K. SOONG, Jian-Lai ZHOU

Microsoft Research Asia
Beijing, China
{yushi,frankkps,jlzhou}@microsoft.com

Abstract

An auto-segmentation based endpointing algorithm for robust ASR is proposed. The algorithm consists of two successive steps: (1) homogeneous segment partitioning and (2) segment clustering. The first step, due to its self-segmentation nature, does not need a noise model, and is applicable to different noises at various SNR's. The dynamic programming based segment partitioning, which can generate more homogeneous segments than individual frames for clustering, yields a more robust VAD mechanism. Experiments are performed on the AURORA2 digit database by comparing the new algorithm with the ETSI standard for DSR. Quantitative assessment of the new algorithm is performed via different evaluation criteria, including: ROC curves, speech/non-speech discrimination, and speech recognition performance.

Index Terms: speech recognition, endpointing, VAD, auto-segmentation.

1. Introduction

Endpointing or voice activity detection (VAD) is a key component in speech recognition systems and how to detect speech in a robust way, especially in noise, is a challenging problem.

Speech recognition in adverse environments often demands a noise reduction scheme working in combination with a good voice activity detector. The non-speech detection algorithm is an important and sensitive part of most of the existing single microphone noise reduction schemes such as Wiener filtering (WF) or spectral subtraction. On the other hand, frame dropping (FD) is a frequently used technique in speech recognition to reduce the number of insertion errors. Speech frames incorrectly labeled as silence causes unrecoverable deletion errors, while silence frames incorrectly labeled as speech could increase the insertion errors.

In 2002, a new standard incorporating noise suppression methods has been approved by the European Telecommunication Standards Institute (ETSI) for feature extraction and distributed speech recognition (DSR). The so-called advanced front-end (AFE) [1] incorporates an energy-based VAD for estimating the noise spectrum in Wiener filtering speech enhancement (WF AFE VAD), and a different VAD for nonspeech frame dropping (FD AFE VAD). AFE VADs outperformed VADs in other standards like AMR2, AMR1, and G.729, which is obtained by Ramírez *et al* [2].

A typical VAD decomposes the input speech signal into frames and decision is made on each frame. They are generally effective in clean conditions but the performance starts to degrade at lower SNR levels. An algorithm trying to alleviate these drawbacks exploiting longer-term information has been proposed in [2] and yields better discrimination with sustained improvements in speech/nonspeech hit rates.

In this paper we propose a new long-term information based endpointing algorithm, where variable segment length is derived first algorithmically, toward improving speech detection robustness in adverse environments and the performance of speech recognition systems. The new algorithm is based on auto segmentation [3]. Its goal is to divide a time series into homogeneous blocks to minimize the segmentation cost via dynamic programming (DP) that is often employed in alignment and model-fitting sequence segmentation algorithms. A variety of signal processing and related problems such as signal detection and characterization, density estimation, cluster analysis, and classification can be viewed as the search for an optimal partition of data given on a time interval. In the proposed method, the segmentation score function is defined as a homogeneity criterion penalized by segmentation complexity. Due to its auto segmentation nature, this step does not need a noise model, and is applicable to different noises and signal-to-noise ratios (SNR's). On the other hand, since it is a DP based procedure, the algorithm provides a graceful performance in finding segmentation boundaries. Then any long-term information can be extracted from each segment and any frame-based decision rules can be used to judge whether a segment belongs to speech or background noise. In this paper, only a very simple clustering approach is used.

The algorithm is evaluated in the context of the AURORA2 experimental framework [4] and AFE softwares [5] [6]. The benefits of this approach are assessed quantitatively by an performance analysis in comparison with AFE VADs, in terms of receiver operating characteristics (ROC) curves, speech/non-speech discrimination, and recognition performance when the VAD is used for an automatic speech recognition system.

2. Homogeneous Frame Partitioning

For a given time interval $I = \{t, t = 1, \ldots, T\}$ which contains T frames of speech signals and a predefined parameter L $(1 \leq L \leq T)$ which represents the total number of segments to be produced, segmentation $\mathcal{S}(T, L)$ is defined as a set of L blocks

$$\mathcal{S}(T, L) = \{S_l, 1 \leq l \leq L\} \tag{1}$$

where each block is a set of frames defined by consecutive time indices $\mathcal{N}_l = \{n_{l-1} + 1, \ldots, n_l\}$ as

$$S_l = \{\vec{\mathbf{x}}_n, n \in \mathcal{N}_l\} \tag{2}$$

satisfying the nonoverlapping and nonskipping conditions $\bigcup_l S_l = I$ and $S_l \bigcap S_{l'} = \emptyset$ if $l \neq l'$. Here $\vec{\mathbf{x}}_n$ is the d-dimensional feature vector associated with frame n, and n_l is the end frame of segment S_l. The segmentation score function, similarly with Bayesian information criterion (BIC) [7], is defined as a homogeneity criterion penalized by segmentation complexity: the number of parameters in segmentation $\mathcal{S}$. The formulation is

$$\begin{aligned} F_{\mathcal{S}}(T, L) &= H_{\mathcal{S}}(T, L) + P_{\mathcal{S}}(T, L) \\ &= \sum_{l=1}^{L} D_l + \lambda_p \#_{\mathcal{S}}(T, L) \log(T) \end{aligned} \tag{3}$$

where $H_{\mathcal{S}}(T,L)$ is the homogeneity criterion of segmentation $\mathcal{S}(T,L)$ and $P_{\mathcal{S}}(T,L)$ is the penalty item. $D_l = D(n_{l-1}+1, n_l)$ is a measure function of homogeneity associated with segment l positioned from frame $n_{l-1}+1$ to n_l. λ_p is the penalty weight. $\#_{\mathcal{S}}(T,L)$ is the number of parameters in segmentation $\mathcal{S}(T,L)$.

In this paper, $D(n_1,n_2)$ is a within-segment distortion which is defined as

$$D(n_1,n_2)=\sum_{n=n_1}^{n_2}[\vec{\mathbf{x}}_n-\vec{\mathbf{C}}(n_1,n_2)]^T[\vec{\mathbf{x}}_n-\vec{\mathbf{C}}(n_1,n_2)] \quad (4)$$

where

$$\vec{\mathbf{C}}(n_1,n_2)=\frac{1}{n_2-n_1+1}\sum_{n=n_1}^{n_2}\vec{\mathbf{x}}_n \quad (5)$$

is the centroid of the segment. Thus the number of parameters in the segmentation $\mathcal{S}(T,L)$ is $\#_{\mathcal{S}}(T,L) = L \times d$. An optimal segmentation $\mathcal{S}^*(T,L^*)$ can be obtained by minimizing $F_{\mathcal{S}}(T,L)$ in Equation (3) over all segment numbers and segment boundaries:

$$\mathcal{S}^*(T,L^*)=\arg\min_{L,|\mathcal{S}|=L} F_{\mathcal{S}}(T,L) \quad (6)$$

Since the segmentation complexity is independent on the positions of segment boundaries when the number of segments is fixed, we can separate the minimization into two successive procedures: first minimize $H_{\mathcal{S}}(T,L)$ over all $\mathcal{S}$ for each L and then find the minimum value of $F_{\mathcal{S}}(T,L)$ over all L.

The minimum of $H_{\mathcal{S}}(T,L)$ can be found through a DP procedure which can be implemented similar to the level building algorithm [8][9], i.e., the lth level has l segments. So given an L, there are total L levels in DP search. The algorithm derives the optimal partition of the first n frames at level l, $H^*(n,l)$, using previously obtained optimal partitions, i.e., those of the first j frames at level $l-1$, $H^*(j,l-1)$. At each level we must consider all possible ending locations $j, l-1 \le j < n$ of the next-to-last segment of the optimal partition. For each putative j, the distortion function $H(n,l)$ is — by the principle of optimality — the distortion of the optimal subpartition prior to j, $H^*(j,l-1)$, plus the distortion of the last segment itself, $D(j+1,n)$. The former was stored at previous level, and the later is a simple evaluation of D. The desired new optimal segmentation corresponds to the minimum over all j.

In implementation the auto segmentation to partition observation vectors into segments, a constrained DP algorithm is adopted. The number of frames in produced segments is limited in the range $[n_a, n_b]$. The lower bound corresponds to the shortest duration that a segment should dwell, and the upper bound is used for computational savings. Two boundary functions on the values of frame index n for a given value of level l are defined as $B_a(l) = n_a l$ and $B_b(l) = n_b l$. They are used to restrict the range of the optimal segmentation of the first n frames at level l to fall within a reasonable set of the (n,l) plane. And the putative ending locations of the next-to-last segment of the optimal path, j, are bounded by $(n - n_b)$ and $(n - n_a)$. Due to the limitation of minimum length of the segments, there should be at most $\lfloor T/n_a \rfloor$ segments or levels to be built. (The symbol $\lfloor x \rfloor$ means the largest integer not greater than x.) More precisely, define $H^*(n,l)$ to be the value of the distortion of the optimal segmentation $\mathcal{S}^*(n,l)$ of the first n frames at level l, for $1 \le n \le T$. The DP algorithm shown in Fig. 1 finds the optimal segmentation $\mathcal{S}^*(T,L^*)$.

3. Segment Clustering

In this section, we propose a simple way to cluster the partitioned segments produced in last section into 2 classes, i.e., speech and noise. The processing is performed at the segment level. First the segment centroids of the whole sentence are sorted according to a factor of a sum of the mean values of time-domain log energy and cross correlation corresponding to pitch in each segment. Both time-domain log energies and cross correlation coefficients are variance normalized at sentence level. Then another auto segmentation at level 2 is performed on the sorted segment centroids to find the optimal boundary to separate speech segments from noise segments. Though the segmentation penalty can also be used in this step to determine whether speech or noise exists in the time interval, we set $\lambda_p = 0$ in our implementation for simplicity since we suppose there are both speech and noise in each sentence.

```
Frame Partitioning Algorithm
1. Start level ($l = 1$)
    • Calculate search range
          $B_a(1) = n_a, \quad B_b(1) = n_b$          (7)
    • Compute
          $H^*(n,l) = \begin{cases} D(1,n), & B_a(1) \le n \le B_b(1) \\ \infty, & \text{else} \end{cases}$          (8)
2. For $l = 2, \ldots, (L = \lfloor T/n_a \rfloor)$, do
    • Calculate search range
          $B_a(l) = n_a l, \quad B_b(l) = n_b l$          (9)
    • For $n = B_a(l), \ldots, B_b(l)$, do
        – Compute
              $H^*(n,l) = \min_j \{H^*(j,l-1)+D(j+1,n)\}$,          (10)
          for $n - n_b \le j \le n - n_a$.
        – The value of $j$ where this minimum occurs is stored
          as $p(n,l)$.
3. Select
          $L^* = \arg\min_l [H^*(T,l) + \lambda_p l d \log(T)]$          (11)
   as the optimal number of segments.
4. Backtrack using $p$ to identify the end locations of individual blocks of the optimal segmentation $\mathcal{S}^*(T,L^*)$ in the following way. Let $n_{L^*} = T$, $n_{L^*-1} = p(n_{L^*}, L^*)$, $n_{L^*-2} = p(n_{L^*-1}, L^*-1)$, etc. Then the last block in $\mathcal{S}^*(T,L^*)$ contains frames $n_{L^*-1}+1, \ldots, n_{L^*} = T$, the next-to-last block in $\mathcal{S}^*(T,L^*)$ contains frames $n_{L^*-2}+1, \ldots, n_{L^*-1}$, and so on.
5. Compute centroid $\vec{\mathbf{C}}_l = \vec{\mathbf{C}}(n_{l-1}+1, n_l)$ for each segment $l, l = 1, \ldots, L^*$.
```

Figure 1: *Frame partitioning algorithm.*

4. Experimental Results

Several experiments can be formed to evaluate the performance of VAD algorithms. The analysis is normally focused on the determination of misclassification errors at different SNR levels, and the influence of the VAD decision on speech processing systems. The experimental framework and the objective performance tests conducted to evaluate the proposed algorithm are described in this section.

In this study, the AURORA2 database and recognizer [4] was used. Features used in auto segmentation were the time-domain log energy, root mean square, cross correlation, and Mel-frequency cepstral coefficients (MFCC's). This kind of feature type is a compromise between small variances of start and end point estimate errors, which was investigated in our preliminary research [10]. Fixed hangovers, therefore, should be effective. Auto segmentation was applied in each 0.5 sec interval. Segment length was limited to 0.03–0.25 sec. The segmentation complexity penalty weight in Equation (3) was set to $\lambda_p = 0.2$;

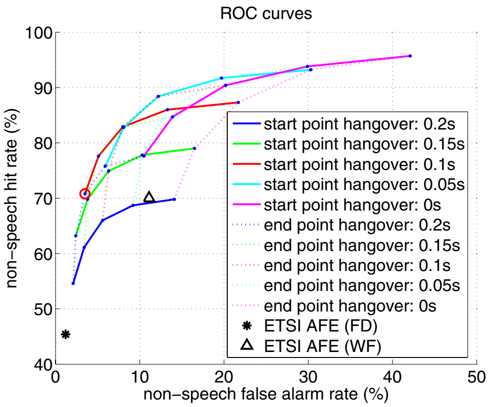


Figure 2: *ROC curves.*

The non-speech hit-rate (HR_0) and the speech hit-rate (HR_1) were defined as the fraction of all actual pause or speech frames that are correctly detected as pause or speech frames, respectively:

$$\mathrm{HR}_0 = \frac{\mathrm{C}(0|0)}{\mathrm{C}_{\mathrm{ref}}(0)}, \quad \mathrm{HR}_1 = \frac{\mathrm{C}(1|1)}{\mathrm{C}_{\mathrm{ref}}(1)} \qquad (12)$$

where $C_{\mathrm{ref}}(0)$ and $C_{\mathrm{ref}}(1)$ are the counts of real non-speech and speech frames in the whole database, respectively, while $C(0|0)$ and $C(1|1)$ are the counts of non-speech and speech frames correctly classified. For the calculation of the false-alarm rate (FAR) as well as the hit rate, the "real" speech frames and "real" speech pauses were determined by aligning clean test data to a set of HMM models trained on clean data from both training and test sets in the database.

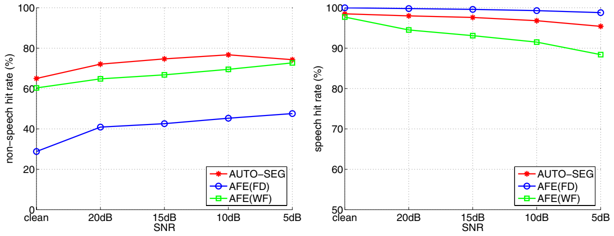
4.1. Receiver operating characteristic curves

We compare speech detection performance by means of the Receiver operating characteristic (ROC) curves that completely describes the VAD error rate.

HR_0 and FAR_0 (=$100 - \mathrm{HR}_1$) were determined in each noise condition for the proposed VAD. HR_0 as a function of FAR_0 for different hangovers is shown in Fig. 2. The results are averaged values over all noise types and SNR levels. The operating point of the AFE VADs [1] are also included. In Fig. 2, each solid line corresponds to one start point hangover, while each dotted line has a same end point hangover. (Hangover is the appended time duration to the time period in which voice activity is detected. It is commonly used in voice activity detector to produce an extended voice detection period.) Different colors represent different hangover values, i.e., color blue, green, red, cyan, and magenta represent hangovers of 0.2, 0.15, 0.1, 0.5, 0.0 sec, respectively. From this figure, we can see that hangovers of 0.1 and 0.2 sec for start and end points (marked as a red circle) is a proper choice, which agrees with the mean and variance analysis in our previous work [10]. Therefore, we chose it as our operating point for latter analysis and experiments. It can be derived from these plots that WF AFE VAD yields high HR_0 but also yields high FAR_0. On the other hand, FD AFE VAD has been planned to be conservative since it is only used in the DSR standard for frame dropping. Thus, it exhibits low HR_0 working on an also low FAR_0. Auto segmentation based VAD yields a much lower FAR_0 and a little bit higher HR_0 than WF AFE VAD. It also yields a much higher HR_0 but also a slightly higher FAR_0 than FD AFE VAD.

4.2. Speech/non-speech discrimination analysis

Second, the proposed VAD was evaluated in terms of its ability to discriminate between speech and background periods at different SNR levels. Detection performance as a function of the SNR was assessed in terms of HR_0 and HR_1.



(a) non-speech hit rate (HR_0) (b) speech hit rate (HR_1)

Figure 3: *Speech/non-speech discrimination analysis.*

Fig. 3 provides the results of this analysis and compares the proposed VAD algorithm to standard AFE VADs in clean conditions and SNR levels ranging from 20 to 5 dBs. Results for the two VADs defined in the AFE DSR standard for estimating the noise spectrum in the Wiener filtering stage and non-speech frame dropping are both provided. The WF AFE VAD yields a poor speech detection performance with a fast decay of HR_1 at low SNR values, while the FD AFE VAD achieves a high HR_1 but mediocre results for non-speech detection. The proposed VAD achieves a better compromise. It behaves well in detecting non-speech as well as exhibits a mild degradation in detecting speech at low SNR's in speech detection.

4.3. Influence of the VAD on a speech recognition system

Although the discrimination analysis and the ROC curves are effective to evaluate the given algorithm, the influence of the VAD decision on the performance of different feature extraction schemes was studied. Recognition results for the auto segmentation based VAD replacing the AFE VADs were provided.

VAD is playing two important roles in speech recognition in adverse environments. In noise reduction, since noise parameters such as its spectrum are updated during non-speech periods, a good VAD algorithm is critical for an effective estimation of noise that is required by speech enhancement systems. On the other hand, non-speech frame dropping is strongly influenced by the performance of the VAD in effectively reducing the number of insertion errors caused by the noise but not leading to too many irrecoverable deletion errors caused by speech misclassification errors. Thus an effective VAD for robust speech recognition needs a compromise between speech and non-speech detection accuracy.

The reference front-end (baseline) is what is used in the ETSI AURORA project for DSR [5]. The AFE features are extracted by means of the ETSI software [6]. The recognizer, published with the AURORA2 database [4], is based on the hidden Markov model toolkit (HTK) software package [11]. We only used clean training in our analysis.

In order to compare the proposed method to the AFE standard, the VADs of the full AFE standard [1] (including both the noise estimation VAD and frame dropping VAD) were replaced by the proposed VAD. Results of the HMM based VAD was also provided as a reference. All results were averaged over the three test sets of the AURORA2 recognition experiments. More clearly, the experiment structure is:

1. incorporate frame dropping to the baseline system
2. replace the WF VAD of the AFE standard and do not perform frame dropping
3. replace the FD VAD of the full AFE standard
4. replace both WF and FD VADs of the full AFE standard

Notice that, particularly, for the last three AFE based experiments,

Table 1: *Influence of VADs on frame dropping incorporated to the baseline.*

System	baseline	baseline + FD		
VAD (FD)	–	ref	AFE	proposed
Clean	99.0	99.2	98.6	99.0
20 dB	94.1	97.4	96.0	97.3
15 dB	85.0	93.7	91.3	93.5
10 dB	65.5	81.5	78.4	82.1
5 dB	38.6	56.7	53.3	59.0
0 dB	17.1	30.4	26.9	30.0
-5 dB	8.5	15.1	12.6	12.0
Avg. (0-20 dB)	60.1	71.9	69.2	72.4

Table 2: *Influence of VADs on noise suppression in AFE.*

System	AFE without FD		
VAD (WF)	ref	AFE	proposed
Clean	99.1	99.1	99.1
20 dB	98.0	98.0	98.0
15 dB	96.6	96.4	96.5
10 dB	92.5	92.3	92.5
5 dB	82.3	82.2	82.2
0 dB	58.2	58.0	57.9
-5 dB	27.5	26.9	27.2
Avg. (0-20 dB)	85.5	85.4	85.4

we have used the same configuration with the standard [1]. The same feature extraction scheme was used for both training and testing. If FD is utilized, only exact speech periods are kept and consequently, all the frames classified by the VAD as non-speech are discarded.

Table 1 to 4 exhibit all recognition results in the clean training. Note that AFE standard uses different VADs for noise suppression and frame dropping. Table 1 shows the word accuracies obtained for the baseline and the modified baseline incorporating the VADs under investigation for frame dropping. We can observe that a 10% word error rate reduction (from 30.8% to 27.6%) is achieved by the proposed VAD in comparison with the FD AFE VAD. Table 2 demonstrates the effectiveness of noise reduction scheme in robust speech recognition. All VADs have achieved improvement of more than 60% on the baseline shown in Table 1. Both the proposed VAD and the WF AFE VAD have a similar performance with the HMM based reference. Table 3 shows the recognition results of the full AFE standard and the modified standard via only replacing the FD VAD by others. The word error rate was reduced from 14.4% to 13.8% when the proposed VAD was used. Table 4 shows the experimental results of the full AFE standard and the modified standard via replacing both WF and FD VADs by others. Another absolute improvement of 0.1% was achieved by the auto segmentation based VAD.

5. Conclusion

In this paper, we propose a robust endpoint detection algorithm based on auto segmentation. Due to the self segmentation nature, the approach does not need any noise models, and is applicable to different noises and SNR's. Since DP based procedure is used, the algorithm provided a graceful performance in finding segmentation boundaries. Though a simple segment clustering method based on segment centroids was used in this paper, any long-term information can be extracted from the segments and any frame-based VAD decision rules can be used. The proposed algorithm was evaluated on the test sets in the Aurora2 database. The proposed VAD outperforms the AFE standard VADs when used for WF, FD, and both of them. The best recognition performance is obtained when the proposed auto segmentation based VAD is used.

Table 3: *Influence of VADs on frame dropping in AFE.*

System	full AFE		
VADs (WF/FD)	AFE/ref	AFE/AFE	AFE/proposed
Clean	99.2	98.8	99.2
20 dB	98.2	97.8	98.0
15 dB	96.8	96.5	96.7
10 dB	93.2	92.5	93.0
5 dB	83.4	82.3	83.3
0 dB	59.9	58.8	60.1
-5 dB	28.4	27.3	28.0
Avg. (0-20 dB)	86.3	85.6	86.2

Table 4: *Influence of VADs on both noise suppression and frame dropping in AFE.*

System	full AFE		
VADs (WF/FD)	ref/ref	AFE/AFE	proposed/proposed
Clean	99.3	98.8	99.1
20 dB	98.2	97.8	98.1
15 dB	96.8	96.5	96.8
10 dB	93.2	92.5	93.1
5 dB	83.5	82.3	83.3
0 dB	60.7	58.8	60.5
-5 dB	30.0	27.3	29.3
Avg. (0-20 dB)	86.5	85.6	86.3

The reduction of the word error rate was 4.4% over AFE VADs. Finally, when comparing the word accuracies to the performance of the recognition system using the HMM-based reference VAD, we can see that the performance of the proposed algorithm is very close to that of the HMM-based reference. In almost all test sets, the proposed VAD algorithm is observed to outperform AFE standard VADs.

6. References

[1] ETSI ES 202 050: "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms".

[2] Ramírez, J., Segura, J. C., Benítez, C., Torre, Á. d. l., Rubio, A., "Efficient voice activity detection algorithms using long-term speech information", Speech Communication, 42 (2004) p 271–287.

[3] Svendsen, T., and Soong, F. K., "On the automatic segmentation of speech signals", ICASSP1987, p 77–80.

[4] Hirsch, H. G., and Pearce, D., "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", ISCA ITRW ASR2000, p 181–188.

[5] ETSI ES 201 108: "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms".

[6] ETSI ES 202 212: "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm".

[7] Chen, S. S., Gopalakrishnan, P. S., "Clustering via the Bayesian information criterion with applications in voice recognition", ICASSP'98, Vol 1, p 645–648.

[8] Myers, C. S., Rabiner, L. R., "A level building dynamic time warping algorithm for connected word recognition", IEEE Trans. ASSP. 29(2):284C297, 1981.

[9] Myers, C. S., Rabiner, L. R., "Connect digit recognition using a level-building DTW algorithm ", IEEE Trans. ASSP. 29(3):351C363, 1981.

[10] Shi, Y., Soong, F. K., and Zhou, J.-L., "Auto-segmentation based partitioning and clustering approach to robust endpointing", ICASSP2006.

[11] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., The HTK Book (for HTK Version 3.1), 2001.