Improved feature extraction based on spectral noise reduction and nonlinear feature normalization

José C. Segura, Javier Ramírez, Carmen Benítez, Ángel de la Torre, Antonio Rubio

Dpto. de Electrónica y Tecn. de Comp. Universidad de Granada, 18071-Granada, SPAIN

{segura,javierrp,carmen,atv,rubio}@ugr.es

Abstract

This paper is mainly focused on showing experimental results of a feature extraction algorithm that combines spectral noise reduction and nonlinear feature normalization. The successfulness of this approach has been shown in a previous work, and in this one, we present several improvements that result in a performance comparable to that of the recently approved AFE for DSR. Noise reduction is now based on a Wiener filter instead of spectral subtraction. The voice activity detection based on the full-band energy has been replaced with a new one using spectral information. Relative improvements of 24.81% and 17.50% over our previous system are obtained for AURORA 2 and 3 respectively. Results for AURORA 2 are not as good as those for the AFE, but for AURORA 3 a relative improvement of 5.27% is obtained.

1. Introduction

In a previous work [1] we have presented a feature extraction algorithm that combines spectral noise reduction and nonlinear feature normalization. In this paper, we present several improvements of the system that yield a performance comparable to that of the AFE [2]. The block diagram of the system is shown in Fig. 1. Main blocks are: voice activity detection (VAD), noise reduction (WF), frame-dropping (FD) and feature normalization (FN).

Voice activity detection is used for the estimation of the mean noise spectrum in the spectral noise reduction block, and for the frame-dropping algorithm at the back-end. The performance of the whole system is greatly affected by this block and therefore, an accurate VAD is needed for high performance. The previous VAD, based on the full-band energy, has been replaced by a more accurate one, which uses detailed spectral information.

Spectral noise reduction has also been improved. In the previous version of the system it was based on a simple spectral subtraction algorithm, now relies on a more sophisticated Wiener-filtering algorithm (WF). The implementation of the noise reduction filter is based on that proposed for the AFE.

This work has been supported by the Spanish Government under the CICYT project TIC2001-3323.

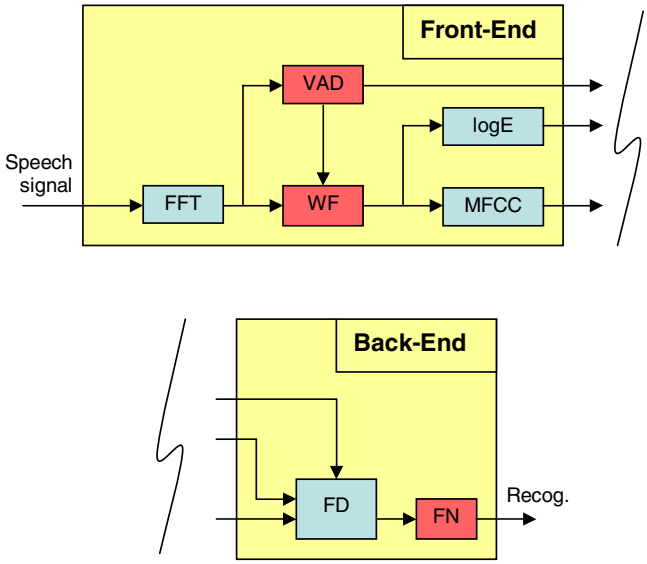

Figure 1: Block diagram of the system

The feature normalization algorithm, previously performed in a sentence-by-sentence basis, has been replaced by a segmental version, which provides effective normalization with delays of about half a second. The performance of the segmental version is comparable to that of the *batch* version.

The organization of the paper is as follows. In section 2, front-end processing is described, including VAD and Wiener filter implementation details. In section 3, we describe the segmental version of the feature normalization algorithm. Experimental results are shown in section 4, and in section 5, we summarize the conclusions of this work.

2. Front-End processing

2.1. LTSE VAD Algorithm

The front-end uses a new VAD that reports important improvements in speech/pause detection accuracy in low SNR noisy conditions. The proposed algorithm is based on the estimation of the averaged maximum SNR over a

neighborhood of the actual frame, and can be described as follows.

During a short initialization period, the mean noise spectrum $N(k)$ (k= 0, 1, ..., *NFFT*-1) is estimated. The input utterance is decomposed into overlapped frames being their spectrum, namely $X(k,n)$, processed by means of a $(2N+1)$-frame window. By defining the so called long-term spectral envelope (LTSE) as

$$LTSE(k) = max\left\{X(k,n+l)\right\}_{l=-N}^{l=+N} \quad (1)$$

where n is the actual frame and k= 0, 1, ..., *NFFT*-1.

The decision rule is formulated in terms of the long-term spectral divergence (LTSD) calculated as the deviation of the LTSE respect to the mean noise spectrum

$$LTSD = 10\log_{10}\left(\frac{1}{NFFT}\sum_{k=0}^{NFFT-1}\frac{LTSE^2(k)}{N^2(k)}\right) \quad (2)$$

which is compared to an adaptive threshold γ.

The threshold is fixed during the initialization of the VAD according to the observed noise energy E. By defining optimal thresholds γ_0 and γ_1 for clean and high noise conditions, respectively, a linear adjust is used. The optimum threshold γ is calculated as a function of the estimated noise energy E

$$\gamma = \begin{cases} \gamma_0 & E \le E_0 \\ \frac{\gamma_0-\gamma_1}{E_0-E_1}E + \gamma_0 - \frac{\gamma_0-\gamma_1}{1-\frac{E_1}{E_0}} & E_0 < E < E_1 \\ \gamma_1 & E \ge E_1 \end{cases} \quad (3)$$

where E_0 and E_1 are the estimated average noise energy for clean and high noise conditions, respectively.

The VAD is defined to be adaptive to time-varying noise environments with the following recursion for updating the noise spectrum during non-speech periods

$$N(k) = \alpha N(k) + (1-\alpha)N_K(k) \quad (4)$$

where N_K is the average spectrum magnitude over a K-frame neighborhood

$$N_K(k) = \frac{1}{2K+1}\sum_{l=-K}^{K} X(k,n-l) \quad (5)$$

and k= 0, 1, ..., *NFFT/2*.

Finally, a hangover was found to be beneficial to maintain a high accuracy detecting speech periods at low SNR levels. Thus, the LTSE VAD yields an excellent classification of speech and pause periods. An example of the operation of the LTSE VAD on an utterance of the Spanish SpeechDat-Car database is shown in Fig. 2.

When compared to the previously published VAD at ICSLP 2002, this new VAD leads to a more efficient application of the noise suppression and frame-dropping algorithms. As an example, for the utterances recorded

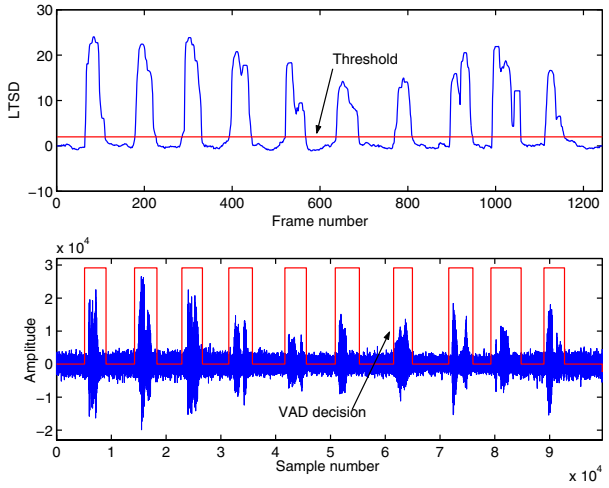

Figure 2: *VAD output for an utterance of the Spanish SpeechDat-Car database (recording conditions: high speed, good road, distant microphone).*

with the distant microphone under high noisy conditions of the Spanish SpeechDat-Car database, the LTSD-based VAD correctly detects a 93.71% of the real speech frames while our previously proposed quantile-based VAD only detected 85.82%. Further improvements are obtained when the SNR becomes lower since the noise rapidly degrades the speech/pause detection accuracy of VAD's that uses the full-band energy as feature for the formulation of the decision rule.

For the experiments conducted on the AURORA-2 database, the 8 kHz input signal sample was decomposed into overlapping frames with a 10-ms shift. A 12-frame long-term window and $NFFT$= 256 was found to be good choices for the noise conditions being studied. Optimal detection threshold γ_0= 5 dB and γ_1= 1.5 dB were determined for clean and noisy conditions, respectively, while the threshold calibration curve was defined between E_0= 30 dB (low noise energy) and E_1= 50 dB (high noise energy). The hangover mechanism delays the speech to non-speech VAD transition during 8 frames while it is deactivated when the LTSD exceeds 30 dB. For the noise update algorithm a forgetting factor α=0.95, and a 3-frame neighborhood (K=3) are used.

2.2. Spectral noise reduction

The Wiener filter is designed in two steps as described in [2]. Only the first stage filter without mel-scale warping is used in this work. Temporal and frequency smoothing is applied to the magnitude spectrum of noisy frames, and the maximum attenuation is fixed at 22dB. A FIR filter is derived from the frequency domain design and further smoothed by truncation of its impulse response with a Hanning window of length 17.

Noise spectrum estimation is performed using a re-

cursive first order filter with a forgetting factor $\lambda = 0.99$. The update is performed for frames labeled as non-speech by the VAD.

3. Back-end processing

Features obtained by the front-end are further processed at the back-end. First, frame dropping is applied to remove long speech pauses from the input feature stream. The algorithm simply discards all input frames labeled as non-speech by the VAD.

After frame dropping, feature normalization is performed. It is based on a nonlinear transformation that maps the estimated probability distribution of each cepstral coefficient into a Gaussian reference one [1, 3, 4]. The transformation is obtained by matching the cumulative distributions functions (CDF) of distorted features C_y with the reference Gaussian distribution C_x

$$C_x(x) = C_y(y) \quad (6)$$

$$x = C_x^{-1}(C_y(y)) \quad (7)$$

In previous works [1, 3], we have approximated this transformation using the cumulative histogram of each distorted feature to obtain an estimation of its CDF. In this work, we have replaced this estimation with a more suitable one for a segmental implementation. The new approach is formulated in terms of the relation between order statistics of a data set and the corresponding CDF [5]. This is a very efficient approach that has been successfully applied to feature normalization in speaker recognition systems [6, 7].

Let Y_t be a temporal buffer for a given distorted feature

$$Y_t = \{y_{-T}, \ldots, y_t, \ldots, y_T\} \quad (8)$$

The order statistics of this data set can be obtained by simply rearranging the data in ascending order

$$y_{(1)} \leq y_{(r)} \leq \cdots \leq y_{(r)} \leq \ldots y_{(2T+1)} \quad (9)$$

An asymptotically unbiased point estimation of the CDF can be obtained as

$$\hat{C}(y_{(r)}) = \frac{r - 0.5}{2T + 1} \qquad \forall\ r = 1, \ldots, 2T + 1 \quad (10)$$

Using (10) and (7), an estimation of the transformed value of y_t can be obtained as

$$\hat{x}_t = C_x^{-1}(\hat{C}(y_t)) = C_x^{-1}\left(\frac{r(y_t) - 0.5}{2T + 1}\right) \quad (11)$$

where $r(y_t)$ denotes the rank of y_t (i.e. the index r of the order statistics that corresponds to the value y_t). This value can be obtained by counting the number of values less or equal than y_t in the temporal buffer Y_t. Note that as C_x and T are fixed, if the values

$$G[r] = C_x^{-1}\left(\frac{r - 0.5}{2T + 1}\right) \quad \forall\ r = 1, \ldots, 2T + 1 \quad (12)$$

are tabulated in advance, the transformed value (11) can be obtained by simply indexing the table G.

This algorithm is much more efficient than HEQ [3] because only $2T$ comparisons are needed to obtain the transformed value of a given feature. Furthermore, experimental results have shown that its performance is almost the same achieved with HEQ.

Aurora 2 Relative Improvement				
	Set A	**Set B**	**Set C**	**Overall**
Multi	16,47%	21,79%	20,70%	**19,44%**
Clean	30,46%	30,59%	28,78%	**30,18%**
Average	**23,46%**	**26,19%**	**24,74%**	**24,81%**

Aurora 3 Relative Improvement					
	Finnish	**Spanish**	**German**	**Danish**	**Average**
Well (x40%)	23,62%	6,57%	19,52%		**16,57%**
Mid (x35%)	20,12%	-8,98%	15,34%		**8,83%**
High (x25%)	52,81%	21,36%	19,19%		**31,12%**
Overall	**29,69%**	**4,82%**	**17,97%**		**17,50%**

Table 1: *Relative improvements over the previous system*

4. Experimental results

Experimental results for the proposed system have been obtained for two different versions of the new feature normalization algorithm. Both versions have been evaluated using re-endpointed versions of the databases as described in [1].

The first version is a *batch* one that uses all the features of a given input utterance to perform the normalization. Comparative results for this version are summarized in table 1, where relative improvements are computed using our previous results [1] as the baseline. The higher performance of the new front-end is mainly due to the more effective spectral noise reduction and the better performance of the new VAD.

Official results with respect to the AFE baseline are shown in Tables 2 and 3. It is clear that for AURORA 2, results are not as good as those obtained for the AFE are, but only small differences can be observed. The averaged word error rate is only 0.32% higher and the relative performance is only a 9.72% worse when compared to AFE. For the AURORA 3 databases, the performance of the new system is better than the corresponding AFE baseline. The new system performs better in well-match and medium-mismatch conditions, and the averaged relative improvement is 5.27%.

We have also evaluated the performance of the proposed segmental version of the feature normalization algorithm for a buffer length of 121 frames (a delay of 600 ms). Results are summarized in Tables 4 and 5.

Although results are not as good as those of the *batch* version, they prove the successfulness of a segmental implementation of the feature normalization algorithm with rather short delays.

Aurora 2 Word Error Rate				
	Set A	Set B	Set C	Overall
Multi	6,16%	6,50%	7,27%	**6,52%**
Clean	12,83%	12,07%	13,63%	**12,69%**
Average	**9,49%**	**9,28%**	**10,45%**	**9,60%**

Aurora 2 Relative Improvement				
	Set A	Set B	Set C	Overall
Multi	-8,59%	-4,21%	-3,86%	**-5,89%**
Clean	-21,13%	-10,50%	-4,46%	**-13,54%**
Average	**-14,86%**	**-7,35%**	**-4,16%**	**-9,72%**

Table 2: *Results for AURORA 2. Batch version*

Aurora 3 Word Error Rate					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	4,14%	3,13%	4,37%	6,01%	**4,41%**
Mid (x35%)	10,60%	6,43%	10,10%	14,31%	**10,36%**
High (x25%)	12,69%	10,20%	8,93%	21,07%	**13,22%**
Overall	**8,54%**	**6,05%**	**7,52%**	**12,68%**	**8,70%**

Aurora 3 Relative Improvement					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	-5,88%	6,85%	10,63%	9,35%	**5,24%**
Mid (x35%)	44,44%	-5,76%	-10,26%	22,69%	**12,78%**
High (x25%)	5,23%	-20,71%	-2,06%	-3,23%	**-5,19%**
Overall	**14,51%**	**-4,45%**	**0,15%**	**10,87%**	**5,27%**

Table 3: *Results for AURORA 3. Batch version*

Aurora 2 Word Error Rate				
	Set A	Set B	Set C	Overall
Multi	6,33%	6,55%	7,51%	**6,65%**
Clean	13,16%	12,04%	13,64%	**12,81%**
Average	**9,74%**	**9,29%**	**10,57%**	**9,73%**

Aurora 2 Relative Improvement				
	Set A	Set B	Set C	Overall
Multi	-12,68%	-7,21%	-8,87%	**-9,73%**
Clean	-28,78%	-14,51%	-9,10%	**-19,14%**
Average	**-20,73%**	**-10,86%**	**-8,99%**	**-14,43%**

Table 4: *Results for AURORA 2. Segmental version*

Aurora 3 Word Error Rate					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	4,14%	3,31%	5,09%	6,68%	**4,80%**
Mid (x35%)	10,60%	6,61%	11,27%	16,86%	**11,34%**
High (x25%)	13,25%	8,99%	10,78%	20,44%	**13,37%**
Overall	**8,68%**	**5,89%**	**8,68%**	**13,68%**	**9,23%**

Aurora 3 Relative Improvement					
	Finnish	Spanish	German	Danish	Average
Well (x40%)	-5,88%	1,49%	-4,09%	-0,75%	**-2,31%**
Mid (x35%)	44,44%	-8,72%	-23,03%	8,91%	**5,40%**
High (x25%)	1,05%	-6,39%	-23,20%	-0,15%	**-7,17%**
Overall	**13,46%**	**-4,05%**	**-15,50%**	**2,78%**	**-0,83%**

Table 5: *Results for AURORA 3. Segmental version*

5. Conclusions

In this work, we have presented several improvements of a feature extraction algorithm based on the combination of spectral noise reduction and nonlinear feature normalization.

A new VAD algorithm is presented that improves both the noise estimation and the frame dropping as it provides better speech/noise discrimination.

The feature normalization algorithm has been modified to improve its computational efficiency and to simplify the implementation of its segmental version. The performance of this new algorithm is only slightly worse than that of the corresponding *batch* version.

Finally, the reported results show that the proposed system performance is comparable to that of the AFE for AURORA 2 and even better for AURORA 3.

6. References

[1] J. Segura, C. Benitez, A. de la Torre, and A. Rubio, "Feature extraction combining spectral noise reduction and cepstral histogram equalization for robust ASR," in *Proc. ICSLP'02*, Denver, Colorado, September 2002, pp. 225–228.

[2] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms." in *ES 202 050 Recommendation*, 2002.

[3] A. de la Torre, J. Segura, C. Benitez, A. Peinado, and A. Rubio, "Non-linear transformations of the feature space for robust speech recognition," in *Proc. ICASSP'02*, Orlando, Florida, May 2002, pp. 401–404.

[4] J. Segura, C. Benitez, A. de la Torre, S. Supont, and A. Rubio, "VTS residual noise compensation," in *Proc. ICASSP'02*, Orlando, Florida, May 2002, pp. 409–412.

[5] R. Suoranta, K.-P. Estola, S. Rantala, and H. Vaataja, "Pdf estimation using order statistic filter bank," in *Proc. of ICASSP'94*, vol. 3, April 1994, pp. 625–628.

[6] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Speaker Oddyssey 2001 conference*, June 2001.

[7] B. Xiang, U. V. Chaudhari, J. Navratil, G. N. Ramaswamy, and R. A. Gopinath, "Short-time gaussianization for robust speaker verification," in *Proc. ICASSP'02*, Orlando, Florida, May 2002, pp. 681–684.