Published in Proceedings of the 3rd COST 276 Workshop on
Information and Knowledge Management for Integrated Media Communication, 120-125, 2002
which should be used for any reference to this work

DISTRIBUTED SPEAKER RECOGNITION USING THE ETSI AURORA STANDARD

S. Grassi, M. Ansorge, F. Pellandini, P.-A. Farine
Institute of Microtechnology (ESPLAB), University of Neuchâtel,
Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland.
Phone: +41 32 718 34 32; Fax: +41 32 718 34 02; e-mail: Sara.Grassi@unine.ch.

***Abstract* – The ETSI "Aurora" is a standard for distributed *speech* recognition over the mobile cellular network. We have investigated the use of the features defined in this standard for *speaker* recognition, in a text-independent system based on Gaussian Mixture Models (GMM). The application context is distributed *speaker* recognition for user authentication on the mobile cellular network. We have found that the use of the Aurora parameters would improve the performance compared with the existing alternative which is performing speaker recognition on GSM coded speech.**

***Index Terms*—ETSI Aurora, Distributed Speaker Recognition.**

1. INTRODUCTION

Automatic speaker recognition is the use of a machine to recognize a person from a spoken phrase [1]. It includes verification and identification. In verification, the machine is used to verify a person's claimed identity from his/her voice, while in identification there is no "a priori" identity claim, and the system decides who the person is. In text-independent speaker recognition, the user can utter an arbitrary phrase, whereas in text-dependent systems a fixed "voice password" is uttered, and in "text-prompted" systems the user is asked to repeat a phrase.

There is a tendency to use the mobile phone to access data and services, which is likely to increase with the advent of Wireless Application Protocol (WAP) and 2.5/3G communication systems. When access control to these data and services is needed, authentication by voice seems the more natural and easily implementable choice for a mobile phone.

With nowadays technology, advanced (speech and speaker) recognition algorithms are too complex to be implemented on a portable phone, mainly due to power consumption constraints. Thus, a distributed (client-server) approach is used. In the straightforward scenario, speech is coded (compressed) in the mobile phone, transmitted over the cellular network, and recognition is performed in the server side either by using the decoded speech, or by deriving features directly from the encoded parameters [2]. In this approach, the recognition performance is degraded by channel transmission errors and distortion introduced by the compression algorithms. Furthermore, due to the coexistence of different compression algorithms in the fixed (circuit- and packet- switched) and mobile telephone networks, it is impossible to predict which combination of coders and channels the speech has undergone before arriving to the server. The consequent mismatch between speech used in training the recognition system and speech to be recognized is another significant source of performance degradation. A solution to these problems is to move the extraction of recognition features (recognition front-end) to the client side, compressing and sending the extracted parameters, through an error protected data channel on the mobile network, to the remote "back-end" recognizer. Thus, the European Telecommunication Standards Institute (ETSI) has standardized a front-end for Distributed *Speech* Recognition (DSR), under the name "Aurora". This standard is briefly explained in Section 2.

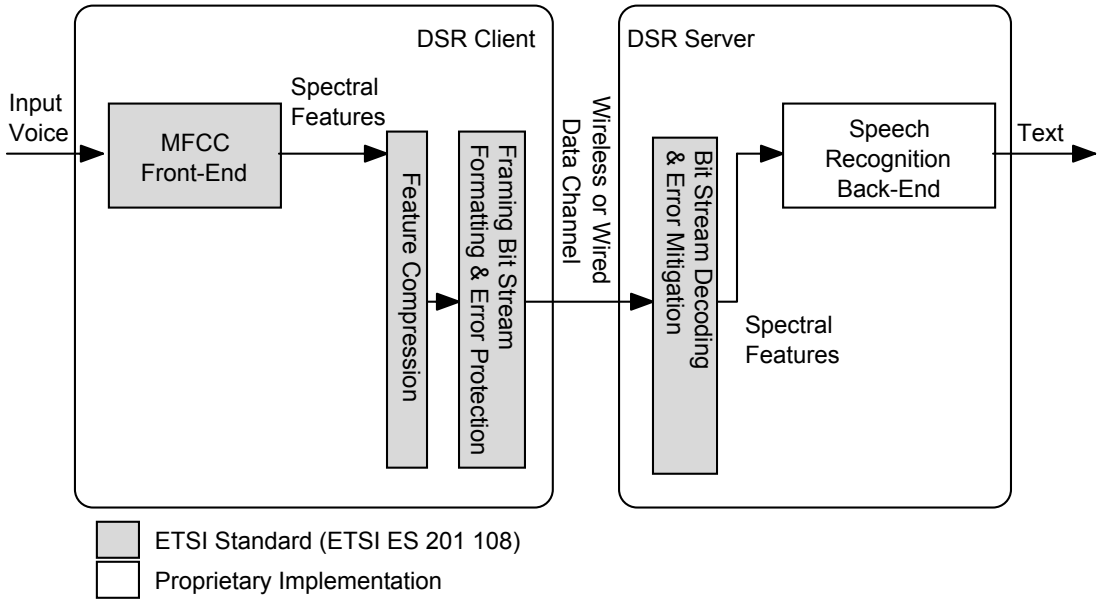

Figure 1: Distributed speech recognition and the ETSI Aurora standard.

In this paper we investigate the use of the ETSI Aurora parameters for *speaker* recognition, using the text independent recognition system described in Section 3. As we wanted to compare the use of the Aurora parameters with the straightforward scenario, namely performing recognition over GSM coded speech, experiments on this topic are reported in Section 4, whereas experiments using the Aurora features are described in Section 5. Finally conclusions and further work are given in Section 6.

2. THE ETSI AURORA STANDARD FOR DISTRIBUTED SPEECH RECOGNITION

In a distributed speech recognition (DSR) architecture the recognizer front-end is located at the terminal (client side) and is connected over a data network to a remote back-end recognition server, as shown in Figure 1 [3]. The ETSI "Aurora" standard for Distributed *Speech* Recognition (ES 201 108) covers front-end feature extraction and compression, as well as bit-stream framing, formatting and decoding, and error protection and mitigation, as shown in Figure 1. Our work deals only with feature extraction and compression, which are detailed in [4] and briefly explained as follows.

a. Feature extraction algorithm [4]

Extracted features are 13 MFCC (Mel-Frequency Cepstral Coefficients), as well as a logarithmic energy measure.

The speech signal is sampled at 8, 11 or 16 kHz and passed through first order offset-compensation and pre-emphasis filters. The resulting signal is segmented into overlapping frames of 25 ms (8 and 16 kHz case) or 23.27 ms (11 kHz case), producing a frame every 10 ms. A Hamming window is applied to each frame, which is then zero padded to 256 samples for the 8 and 11 kHz sampling rate, and 512 samples for 16 kHz. FFT and magnitude spectrum are computed on these frames. Mel-filtering is performed as follows. The useful frequency range [64 Hz, fs/2] is divided into 23-channels that are equidistant in the mel frequency domain. A triangular, half-overlapped window is used to calculate the weighted sum of the FFT magnitude spectrum values in each band. The natural logarithm of the mel filtering output is calculated. The obtained values are transformed to 13 cepstral coefficients,

c0 - c12, using Discrete Cosine Transform (DCT). The c1 – c12 MFCC are a spectral envelope measure, whereas c0 is an energy measure. An alternative energy measure (denoted as logE) is added to the features, calculated as the logarithm of the energy of each frame after offset-compensation, but before pre-emphasis.

b. Feature compression algorithm [4]

The feature vector $y_m = [c_1 - c_{12} \quad c_0 \quad \log E]$ is compressed using Split Vector Quantization (SVQ). Features are grouped into pairs, and each pair is quantized using its own codebook, achieving a data rate of 4800 bps.

3. THE SPEAKER RECOGNITION SYSTEM

The speaker recognition system used in all the experiments is based on Gaussian Mixture Models (GMM) classifiers [5]. A number of N=16 mixtures was used and gaussian densities were represented by diagonal covariance matrices. This system was programmed in Matlab, using H2M [6]. Feature extraction varies for the different experiments, as explained in Sections 4 and 5.

For speaker identification, given a sequence of feature vectors from an unknown speaker signal, the recognized speaker is obtained with the maximum likelihood decision rule. For speaker verification, a world model is constructed to normalize the scores, which are then compared to a threshold in order to accept or reject the speaker.

For all the experiments, the same features are used for training and testing (matching condition).

a. Speaker recognition protocol on TIMIT

The TIMIT database [8] contains speech from 630 speakers, each of them speaking 10 phonetically-rich sentences. The speech signal is recorded at 16 kHz. The text material consists of 2342 sentences, divided into 2 dialect sentences (SA sentences), 450 phonetically compact sentences (SX sentences) and 1890 phonetically diverse sentences (SI sentences). Each speaker reads the two SA sentences, 5 of the SX sentences and 3 of the SI sentences.

We used the “long training / short test” protocol [7] for speaker recognition on the TIMIT database. The features corresponding to the 5 SX sentences are concatenated for training each speaker model. 430 speakers of the database (147 women and 283 men) are used in the speaker identification system for testing. The two SA and the three SI sentences of every speaker are tested separately (430x5=2150 test patterns of 3.2 seconds each, in average). The experiments are totally text independent (SA sentences are used in the test set).

The remaining 200 speakers of the database are used to train the world model needed for the speaker verification experiments. 2150 client accesses and 2150 impostor accesses are made (for each client access, an impostor speaker is randomly chosen among the 429 remaining speakers).

4. EXPERIMENTS ON GSM CODED SPEECH

For comparison reasons, we report results of previous experiments [2] on speaker recognition over GSM coded speech, using the speech recognition system of Section 3.

The whole TIMIT database was downsampled from 16 kHz to 8 kHz. The 16 kHz and the 8 kHz databases are referred to as TIMIT16k and TIMIT8k, respectively. TIMIT8k was

coded / decoded with the GSM EFR speech coder [9], using the C-code implementation provided by ETSI.

The following features are used for recognition: 16 cepstral coefficients (c0-c15) calculated from the speech signal, using DFT based real cepstrum, with a 30 ms frame length and a 10 ms frame rate. Table 1 shows the identification and verification errors obtained with the speaker recognition system on TIMIT16k, TIMIT8k, and the EFR transcoded TIMIT.

We also studied the possibility of performing recognition using features derived directly from the EFR coder parameters rather than from resynthesized speech [2]. The best result (reported in Table 1 was obtained using quantized LPC converted to LSP parameters, ω1-ω10, as well as the logarithm of the energy, calculated from the reconstructed residual.

	Identification Error [%]	*Verification EER* [%]
(1) TIMIT16k	2.2 %	1.1 %
(2) TIMIT8k	13.1 %	5.1 %
(3) EFR TIMIT	28.2 %	6.6 %
(4) Features from EFR encoded parameters	29.3 %	6.7 %

Table 1: Speaker identification error and verification EER (equal error rate) using original TIMIT, downsampled TIMIT, EFR transcoded TIMIT, and features derived from EFR encoded parameters.

5. EXPERIMENTS USING THE ETSI AURORA FEATURES

Speaker recognition experiments were performed using the ETSI Aurora parameters and the recognition system described in Section 3. Table 2 shows the obtained identification and verification errors.

	Non - Compressed			*Compressed*		
Features / Fs	*8 kHz*	*11 kHz*	*16 kHz*	*8 kHz*	*11 kHz*	*16 kHz*
(1) c1-c12	20.0 %	10.3 %	5.7 %	21.9 %	14.1 %	7.7 %
(2) c0 + c1-c12	16.6 %	7.1 %	4.8 %	17.5 %	10.7 %	5.0 %
(3) logE + c1-c12	16.3 %	7.3 %	4.1 %	17.3 %	10.3 %	5.3 %
(4) logE + c0 + c1-c12	18.1 %	9.0 %	5.3 %	19.4 %	11.3 %	6.4 %

Table 2a: Error percentage obtained for speaker identification.

	Non - Compressed			*Compressed*		
Features / Fs	*8 kHz*	*11 kHz*	*16 kHz*	*8 kHz*	*11 kHz*	*16 kHz*
(1) c1-c12	5.60 %	3.81 %	2.37 %	5.88 %	3.70 %	2.33 %
(2) c0 + c1-c12	4.40 %	2.72 %	1.98 %	4.40 %	3.19 %	1.58 %
(3) logE + c1-c12	4.16 %	3.00 %	1.95 %	4.47 %	3.28 %	1.81 %
(4) logE + c0 + c1-c12	4.44 %	3.16 %	2.14 %	4.26 %	2.70 %	2.23 %

Table 2b: Equal Error Rate (EER) percentage obtained for speaker verification.

The TIMIT database was downsampled from 16 kHz to 11 kHz and from 16 kHz to 8 kHz. The Aurora features were extracted from the original TIMIT and from the downsampled databases, using the C-program provided by ETSI, specifying as input parameter the sampling frequency according to the used database. We first tried the c1-c12 alone (corresponding to spectral envelope information) and no energy measure. Then we used c1-c12 with c0 (energy information), c1-c12 with logE (alternative energy information), and c1-c12 with both c0 and logE. These variants were studied for the 8 kHz, 11 kHz, and 16 kHz sampling frequencies.

First, we performed the experiments with non-compressed features, mainly to allow assessment of the degradation introduced by feature compression, as this case would not occur in practice. Then, we repeated the experiments using compressed features. The degradation due to compression is in the range of 0.2-3.8% (average 1.9%) for identification. In the case of verification, results are not conclusive, as in some cases the performance slightly improves.

The best performance is obtained by using c0 + c1-c12 at 16 kHz. Note that there is no advantage in simultaneously using c0 and logE in the feature vector, as this actually decreases the performance.

We observe in Table 1 and 2 that a main source of degradation is in reducing bandwidth. Unfortunately the audio hardware of current mobile phones uses 8 kHz sampling frequency, but this is expected to change with the arrival of cellular phones adapted to the new ETSI Wideband Adaptive Multirate (WB-AMR) speech coding standard [9] which uses speech sampled at 16 kHz.

By comparing cases (1) and (2) in Table 1 respectively with the non-compressed 8 and 16 kHz cases in Table 2, we observe worse performance with the Aurora parameters, probably due to the fact that 13 coefficients are used instead of 16. Note that the choice of the number of cepstral coefficients in the Aurora standard is limited by the bit-rate constraint.

Finally, if we compare cases (3) and (4) in Table 1 with the best 8 kHz compressed result in Table 2, we observe that the use of the ETSI Aurora parameters definitely improves performance, compared with the alternative scenario of performing recognition over GSM coded speech: performance improves by 11 % (from 28.2 % to 17.3 %) in identification and 2 % (from 6.6 % to 4.47 %) in verification.

To the extent of our knowledge the only similar work reported in scientific literature is found in [10]. They use a text-prompted speaker verification system, and test it using Aurora features and the YOHO database, obtaining an EER of 1.22 % in the compressed case, and concluding that Aurora features originally designed for *speech* recognition also work well for *speaker* recognition. Note that they did not specify the used sampling frequency. The best compressed result we obtained in verification is comparable to theirs, especially considering that a text-prompted system is bound to yield better performance than a text-independent system.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the use of the ETSI Aurora parameters (originally designed for distributed *speech* recognition) for *speaker* recognition, using a text-independent system based on Gaussian Mixture Models (GMM).

We found that the use of the Aurora parameters would definitely improve performance in a distributed speaker recognition system, compared with the alternative scenario of performing recognition over coded speech. Experiments took only into account the degradation due to speech coding. A more extensive comparison should include different error channel conditions, to test the effect of the error protection of the Aurora standard with respect to the error protection of the GSM speech channel. In a real situation, the Aurora parameters are expected to perform even better than features extracted from GSM coded speech, due to the mismatch in the training of the recognition system inherent of the latter scenario.

Future and ongoing work includes comparing the performance of the Aurora parameters versus the new WB-AMR ETSI speech coder standard.

A larger and very ambitious goal would be the inclusion of the (speech and speaker) recognizibility constraint into the design of speech coders, which are traditionally designed to

keep listening quality of the reconstructed speech. A small step in this direction may be given by the current Aurora standardization activities in front-end extension for speech reconstruction [11].

7. ACKNOWLEDGEMENTS

This work was partly supported by the Swiss Federal Office for Education and Science under Grant OFES C00.0105 (COST 276 project).

References

[1] J.P., Jr. Campbell, "Speaker Recognition: a Tutorial", Proc. of the IEEE, Vol. 85, no. 9, pp. 1437 –1462, Sept. 1997.

[2] S. Grassi, L. Besacier, A. Dufaux, M. Ansorge, F. Pellandini, "Influence of GSM Speech Coding on the Performance of Text-Independent Speaker Recognition", Proc. of the European Signal Processing Conference 2000, EUSIPCO 2000, Vol. 1, pp. 437-440, Tampere, Finland, Sept. 2000.

[3] D. Pearce, "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed, Speech Recognition Front-ends", in proc. American Voice I/O Society (AVIOS) , San-Jose CA, USA, May 2000.
In http://www.etsi.org/technicalactiv/DSR/Avios%20DSR%20paper.pdf (Sept. 2002).

[4] ETSI ES 201 108 V1.1.2 (2000-04), "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", April 2000. In http://www.etsi.org (Sept. 2002).

[5] D. Reynolds, "Speaker Identification and Verification Using Gaussian Mixture Speaker Models", Proc. of Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland, April 1994, pp. 27-30.

[6] O. Cappé, "H2M : A set of MATLAB/OCTAVE functions for the EM estimation of mixtures and hidden Markov models", in http://tsi.enst.fr/~cappe/mfiles/h2m.tgz (Sept. 2002).

[7] F. Bimbot et al., "Second-order Statistical Methods for Text-Independent Speaker Identification", Speech Communication, Vol. 17, no.1-2, pp. 177-192, Aug. 1995.

[8] Fisher, W. M., V. Zue, J. Bernstein, and D. S. Pallett, "An acoustic-phonetic database", J. Acoust. Soc. Am., Suppl. 1, Vol. 81, p. S92, 1987.

[9] http://www.etsi.org (Sept. 2002).

[10] C. Broun, W. Campbell, D. Pearce, and H. Kelleher, "Distributed Speaker Recognition using the ETSI Distributed Speech Recognition Standard", 2001: A Speaker Odyssey, The Speaker Recognition Workshop, June 18-22, 2001, Crete, Greece.

[11] AU/335/01, ETSI DSR Applications and Protocols Working Group, "New Aurora Activity for Standardization of a Front-End Extension for Tonal Language Recognition and Speech Reconstruction", June 2001.
In http://www.etsi.org/technicalactiv/DSR/Au33501_DSR_reconstruction.pdf
(Sept. 2002).