Published in 10th European Signal Processing Conference EUSIPCO, 437-440, 2000
which should be used for any reference to this work

INFLUENCE OF GSM SPEECH CODING ON THE PERFORMANCE OF TEXT-INDEPENDENT SPEAKER RECOGNITION

S. Grassi[1], L. Besacier[2], A. Dufaux[1], M. Ansorge[1], and F. Pellandini[1]

[1] Institute of Microtechnology, University of Neuchâtel, Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland
Phone: +41 32 7183432; Fax: +41 32 7183402; Email: Sara.Grassi@imt.unine.ch

[2] CLIPS Lab., GEOD team, University Joseph Fourier, BP 53, 38041 Grenoble, Cedex 9, France

ABSTRACT

We have investigated the influence of GSM speech coding in the performance of a text-independent speaker recognition system based on Gaussian Mixture Models (GMM).

The performance degradation due to the utilization of the three GSM speech coders was assessed, using three transcoded databases, obtained by passing the TIMIT through each GSM coder / decoder. The recognition performance was also assessed using the original TIMIT and its 8 kHz downsampled version. Then, different experiments were carried out in order to explore feature calculation directly from the GSM EFR encoded parameters and to measure the degradation introduced by different aspects of the coder.

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 voice, while in identification there is no "a priori" identity claim, and the system decides who the person is. Speaker recognition has applications such as banking over telephone network and security control for confidential information. Due to the increasing demand for mobile communications, it is expected that in the near future many of these transactions will take place through the mobile cellular network.

There exists three GSM Speech coders[1], namely the full rate (FR), half rate (HR) and enhanced full rate (EFR) coder [2]. They compress the speech signal before its transmission, reducing the number of bits needed, while keeping an acceptable quality of the decoded output. Thus, these coders are likely to have an influence on voice recognition performance, together with other perturbations introduced by the mobile cellular network (channel errors, background noise).

In this paper, we investigate the influence of speech coding on speaker recognition performance. Two different experiments are presented. In the first experiment (reported from [3]) the recognition performance degradation due to the utilization of the three GSM speech coders is assessed. In the second experiment, the features for the speaker recognition system are calculated directly from the GSM EFR encoded bit stream. This allows a measurement of the degradation introduced by the different aspects of the coder, and gives guidelines for a better use of the information available in the bit stream, for speaker recognition purposes. The paper is organized as follows. The three GSM speech coders and the GSM transcoded databases are explained in Section 2. The speaker recognition system used in all the experiments is presented in Section 3. Experiments using original and transcoded speech are given in Section 4, whereas experiments using features extracted directly from the GSM EFR encoded parameters are given in Section 5. Finally, conclusions and future work are drawn in Section 6.

2. GSM SPEECH CODERS AND TRANSCODED DATABASES

The three GSM coders work on a 13 bit uniform PCM speech signal, sampled at 8 kHz, which is processed on a frame-by-frame basis, using a 20 ms frame.

The Full Rate (FR) coder, described in GSM 06.10 [2], is a 13 kbps RPE-LTP (Regular Pulse Excitation-Long Term Prediction) coder. A public domain bit exact C-code implementation of the coder is available [4]. The Half Rate (HR) coder is a 5.6 kbps VSELP (Vector Sum Excited Linear Prediction) coder. Its measured output speech quality is comparable to the quality of the FR coder in all tested conditions, except for tandem and background noise conditions. The normative GSM 06.06 [2] gives the bit-exact ANSI-C code for this algorithm. The Enhanced Full Rate (EFR) coder provides substantial quality improvement compared to the FR. This 12.2 kbps coder is based on Algebraic Code Excited Linear Prediction (ACELP). Its bit exact ANSI-C implementation is given in GSM 06.53 [2]. Spectral analysis in the EFR coder is performed once per frame, as explained in Section 2.1.

The whole TIMIT database [5] was downsampled from 16 kHz to 8 kHz, using high quality filtering [3] preserving basically all the frequencies in the range 0-4 kHz. The 16 kHz and the 8 kHz databases will be referred to as TIMIT16k and TIMIT8k respectively. TIMIT8k was coded / decoded with the three GSM coders, using the public domain C-code of the FR coder, and the ETSI ANSI-C code of the HR and the EFR.

[1] Recently, another speech coder, named the Adaptive Multirate (AMR) coder, was standardized by ETSI.

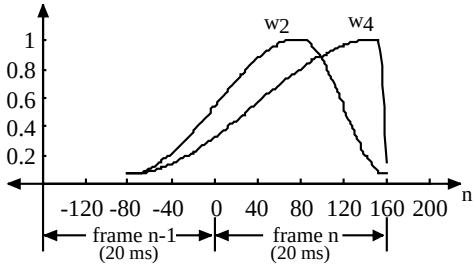

Figure 1: LPC analysis windows in the EFR coder.

Original		*GSM Transcoded*		
TIMIT16k	*TIMIT8k*	*FR*	*HR*	*EFR*
2.2%	13.1%	31.5%	38.5%	28.2%

Table 1: Speaker identification results (% errors).

Original		*GSM Transcoded*		
TIMIT16k	*TIMIT8k*	*FR*	*HR*	*EFR*
1.1%	5.1%	7.3%	7.8%	6.6%

Table 2: Speaker verification results (% EER).

2.1 Spectral Analysis in the GSM EFR Coder

The input speech signal is first high pass filtered. Linear Predictive analysis (LPC) is performed twice per speech frame using autocorrelation and Levinson-Durbin recursion, on the same set of speech samples, with the two different 30 ms asymmetric windows, w_2 and w_4, shown in Figure 1. The two resulting sets of LPC coefficients are converted to two sets of Line Spectrum Pairs (LSP) for quantization and interpolation. The interpolated LSP vectors are reconverted to LPC, obtaining a different LPC filter for each subframe, which is used for calculation of other encoded parameters (LTP lags and gain, and stochastic pulses and gain).

3. SPEAKER RECOGNITION SYSTEM

3.1 System description

The speaker recognition system is based on Gaussian Mixture Models (GMM) classifiers [6]. A number of N=16 mixtures was used and gaussian densities were represented by diagonal covariance matrices. The speaker recognition system was programmed in Matlab, using h2m [7]. 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.

3.2 Speaker recognition protocol on TIMIT

We used the "long training / short test" protocol [8] for speaker recognition on TIMIT. 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).

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

4. EXPERIMENTS USING THE TIMIT AND TRANSCODED DATABASES

The speech analysis module extracts 16 cepstral coefficients (c_0-c_{15}) from the speech signal, using DFT based real cepstrum [9], with a 30 ms frame length and a 10 ms frame rate. Tables 1 and 2 show the identification and verification errors obtained with the speaker recognition system on TIMIT16k, TIMIT8k, and GSM transcoded TIMIT.

A significant performance degradation is observed when using GSM transcoded databases, compared to the normal and downsampled versions of TIMIT. The results are in correspondence with the perceptual speech quality of each coder. That is, the higher the speech quality is, the higher the measured recognition performance. The degradation of the performance is less important for verification than for identification, but is still significant. These results are similar to those obtained in [10], whereas [11] suggests that the GSM coding does not introduce major degradations.

5. EXPERIMENTS USING THE GSM EFR ENCODED PARAMETERS

As we consider that the performance achieved using transcoded speech is not sufficient in a practical context, in this section we investigate the source of the degradation for the EFR coder, as well as the possibility of performing recognition using directly coder parameters rather than parameters extracted from resynthesized speech. Similar experiments using the FR coder are reported in [3]. Results are given in Tables 3 and 4. In Table 3, line (1) corresponds to the baseline (TIMIT EFR experiment reported from Tables 1 and 2).

When extracting features from encoded parameters, we have a frame rate (imposed by the EFR coder) of 20 ms. Thus, to have a meaningful comparison, we have repeated the baseline experiment, but using a frame rate of 20 ms instead of 10 ms. This result is reported in line (2) of Table 3.

Coefficients	*id. error*	*EER*
(1) Baseline: resynthesized speech EFR	**28.2 %**	**6.6 %**
(2) Modified Baseline (20 ms frame rate)	**30.4 %**	**6.7 %**
(3) LPC10 $\rightarrow$ c0-c15	25.5 %	6.3 %
(4) LPC10 $\rightarrow$ c1-c15	31.4 %	6.7 %
(5) LPC12 $\rightarrow$ c0-c15	23.4 %	6.1 %
(6) LPC10 $\rightarrow$ ω1-ω10	32.7 %	7.1 %
(7) LPC10 $\rightarrow$ c1-c10	34.1 %	7.4 %
(8) EFR (no q) $\rightarrow$ c1-c15	33.3 %	7.1 %
(9) EFR (no q) $\rightarrow$ c1-c16	33.2 %	7.3 %
(10) EFR (no q) $\rightarrow$ c1-c20	35.6 %	7.3 %
(11) EFR (with q) $\rightarrow$ c1-c15	35.9 %	7.1 %
(12) EFR (with q) $\rightarrow$ c1-c15 + $\tilde{c}0$	31.5 %	7.2 %
(13) EFR (with q) $\rightarrow$ ω1-ω10	34.5 %	7.0 %
(14) EFR (with q) $\rightarrow$ ω1-ω10 + $\tilde{c}0$	**29.3 %**	**6.7 %**

Table 3: Speaker verification and identification results for the experiments using the GSM EFR encoded parameters.

In the EFR coder, two LPC sets are calculated every 20 ms, thus, three possibilities were considered for each type of feature extraction:

(a) Features calculated using window w_2.

(b) Features calculated using window w_4.

(c) Two sets of features per 20 ms frame, calculated using windows w_2 and w_4.

Only the results concerning (a) are reported in Table 3. Results concerning (b) and (c) can be found in [12].

All the experiences, lines (3) to (14), were carried out using TIMIT8k, but the feature extraction was made compatible with the spectral analysis of the EFR coder (see Section 2.1). LPC coefficients are converted to cepstral coefficients c1-cn using the recursion for minimum phase signals [9]. The cepstral coefficient c0 (energy term) is calculated using log (E), where E is the energy of the LPC residual. When E is not available (features calculated from the coder parameters) the energy term is calculated as $\tilde{c}0 = \log(\hat{E})$, where $\hat{E}$ is the energy of the reconstructed LPC residual. Conversion from LPC to LSP is done using the Matlab function poly2lsf. For lines (3) to (7) the feature extraction is done with a C-program, using double-precision floating-point arithmetic:

(3) Uses c1-c15, from 10-th order LPC, and energy term c0.

(4) Uses c1-c15 from 10-th order LPC, no energy term.

(5) Uses c1-c15 from 12-th order LPC, and c0.

(6) Uses LSP, ω1-ω10, from 10-th order LPC.

(7) Uses 10 cepstral coefficients, c1-c10, from 10-th order LPC.

Feature extraction for lines (8) to (14), is done from the ETSI EFR C-program, which uses a simulated 16-bit fixed-point arithmetic:

(8) Uses c1-c15, from unquantized LPC.

(9) Uses c1-c16, from unquantized LPC.

(10) Uses c1-c20, from unquantized LPC.

(11) Uses c1-c15, from quantized (coded/decoded) LPC.

(12) Uses c1-c15, from quantized LPC, and $\tilde{c}0$, from the energy of the reconstructed LPC residual.

(13) Uses LSP, ω1-ω10, from quantized LPC.

(14) Uses LSP, ω1-ω10, from quantized LPC, and $\tilde{c}0$.

5.1 Comments on Comparisons on Table 3

(1)-(2): The frame rate imposed by the EFR coder (20 ms instead of 10 ms) gives slightly worse results. But the amount of feature vectors is halved.

(3)-(4): The use of c0 is crucial for good performance.

(3)-(5): Increasing LPC order from 10 to 12 improves the performance by a modest amount (~2% on identification) compared with the improvement obtained passing from LPC8 to LPC12 in the FR coder (~8 %) [3].

(4)-(6): The use of LSP ω1-ω10 instead of c1-c15 slightly degrades the performance, but the dimension of the vectors is decreased from 15 to 10.

(6)-(7): Use of LSP gives better results than cepstral coefficients, for the same dimension, when using unquantized LPC.

(11)-(13): Better performance is achieved using LSP ω1-ω10 compared with c1-c15, when using quantized LPC, in spite of the dimension reduction from 15 to 10. This positive result may be due to the fact that the EFR coder does LPC quantization in the LSP domain.

(8)-(9)-(10): Increasing the number of cepstral coefficients beyond c15 does not significantly improve, and may actually degrade, the performance.

(4)-(7): Reducing the number of cepstral coefficients from 15 to 10 decreases the performance.

(4)-(8): Calculations using 16-bit fixed-point arithmetic in the EFR coder decrease the performance.

(8)-(11): LPC quantization decreases the performance.

(11)-(12) & (13)-(14): $\tilde{c}0$ calculated from the reconstructed residual improves the performance.

(2)-(14): Using feature extraction directly from encoded parameters rather than resynthesized speech, improves the performance, for the same frame rate (20 ms).

5.2 Use of Higher Order LPC

In Table 3 it is observed that increasing the LPC order improves the performance, but only 10-th order LPC is available in the EFR encoded parameters. Different experiments we have carried out let us assume that higher order LPC information “leaks” in other encoded parameters (LTP lags and gain, and stochastic pulses and gain) and is thus available in the decoded speech, improving recognition. We investigated the use of this higher order LPC information. The goal is to improve upon (14) in Table 3, the best

result obtained using encoded parameters. Results are given in Table 4. Feature extraction is explained as follows:

(1) LPC from encoded parameters is converted to reflection coefficients k1-k10 and concatenated with reflection coefficients k11-k12 calculated from decoded speech. These concatenated k1-k12 are converted to LSP ω1-ω12, and used as features, together with $\tilde{c}0$ calculated from encoded parameters.

(2) Uses ω1-ω12 calculated from decoded speech, and $\tilde{c}0$ from encoded parameters.

(3) Uses ω1-ω12, calculated from decoded speech, and c0 from decoded speech.

(4) For comparison purposes: Uses ω1 - ω12 from original (TIMIT8k) speech, and c0 from original speech.

In Table 4 it is observed that best results are obtained by using information extracted from encoded parameters, rather than from decoded speech. Naturally, the performance is still better when extracting features from the original speech. We have improved upon line (14) in Table 3, got close to the baseline for speaker identification and improved upon the baseline for verification.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have investigated the influence of GSM speech coding on a text-independent speaker recognition system based on GMM classifiers. The recognition performance when extracting features from GSM transcoded speech was measured, and it was found that is not acceptable for practical applications. Thus, different experiments were carried out, using the EFR coder, to measure the degradation in performance introduced by different aspects of the coder, and to explore the possibility of performing recognition by extracting features directly from coder parameters rather than from transcoded speech.

It was found that the performance can be improved by using feature extraction directly from encoded parameters rather than from transcoded speech, for the same frame rate (20 ms). The performance is also improved by using LSP parameters instead of cepstral coefficients. The best result we have obtained (line 14 in Table 3) is slightly worse than the baseline in performance (line 1 in Table 3), but computationally more efficient (amount of feature vectors is halved, and vector dimension is reduced from 16 to 11).

Future work should include finding ways of improving the baseline, varying either the speaker recognition system, or the feature extraction. For the latter, we would like to explore the use of mel-cepstral coefficients and of LSP weighting functions to emphasize formant structure and attenuate broad-bandwidth components that introduce undesired variability due to environmental factors. When extracting features from the encoded parameters, it was found that the performance can be enhanced by the contribution of the residual (reconstructed from encoded parameters other than LPC). In our experiences this contribution was taken into account by using the energy of the reconstructed residual ($\tilde{c}0$), and higher order LPC information from resynthesized speech. Possible direction of future work is to find effective means to parameterize encoded parameters other than LPC in order to improve recognition performance.

Coefficients	*id. error*	*EER*
(1) k1-k10 from encoded parameters, k11-12 from decoded speech → ω1-ω12 + $\tilde{c}0$	29.2 %	6.1 %
(2) k1-12 from decoded speech → ω1-ω12 + $\tilde{c}0$	29.8 %	6.9 %
(3) k1-12 from decoded speech → ω1-ω12 + c0	32.0 %	6.6 %
(4) k1-k12 from original speech → ω1-ω12 + c0	24.7 %	5.9 %

Table 4: Speaker verification and identification results for the experiment on the use of higher order LPC.

7. ACKNOWLEDGEMENTS

This work was partially supported by the Swiss National Science Foundation under Grant FN 20-53’843, and by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 project).

8. REFERENCES

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

[2] http://www.etsi.org

[3] L. Besacier, S. Grassi, A. Dufaux, M. Ansorge, F. Pellandini, "GSM Speech Coding and Speaker Recognition", Proc. of ICASSP'00, Istanbul, Turkey, June 2000.

[4] http://kbs.cs.tu-berlin.de/~jutta/toast.html

[5] W. Fisher et al., "An acoustic-phonetic database", J. Acoust. Soc. Am., Suppl. 1, Vol. 81, 1987.

[6] 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.

[7] O. Cappé, "h2m: A Set of MATLAB Functions for the EM Estimation of Hidden Markov Models with Gaussian State-conditional Distributions", ENST, Paris, France: http://sig.enst.fr/~cappe/h2m/index.html

[8] 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.

[9] J. R. Deller, J.H. L. Hansen and J. G. Proakis, "Discrete-Time Processing of Speech Signals", New York: Macmillan, 1993.

[10] T.F. Quatieri, E. Singer, R.B. Dunn, D. A. Reynolds, J.P. Campbell, "Speaker and Language Recognition Using Speech Codec Parameters", Proc. of Eurospeech’99, Vol. 2, 1999, pp. 787-790.

[11] M. Kuitert and L. Boves, "Speaker Verification with GSM Coded Telephone Speech", Proc. of Eurospeech’97, Vol.2, 1997, pp. 975-978.

[12] S. Grassi, A. Dufaux, L. Besacier, M. Ansorge, F. Pellandini, "Speaker Recognition on Compressed Speech", Proc. of COST 254 Workshop on Friendly Exchanging Through the Net, Bordeaux, March 2000, pp. 117-122.