AN SVD-BASED SCHEME FOR MFCC COMPRESSION IN DISTRIBUTED SPEECH RECOGNITION SYSTEM

Azzedine Touazi and Mohamed Debyeche

University of Science and Technology Houari Boumediene, Faculty of Electronics and Computer
Signal Processing and Speech Communication Laboratory, BP 32 el Alia
Bab Ezzouar, 16111, Algiers, Algeria
email: {touazi.azzedine, mdebyeche}@gmail.com

ABSTRACT

This paper proposes a new scheme for low bit-rate source coding of Mel Frequency Cepstral Coefficients (MFCCs) in Distributed Speech Recognition (DSR) system. The method uses the compressed ETSI Advanced Front-End (ETSI-AFE) features factorized into SVD components. By investigating the correlation property between successive MFCC frames, the odd ones are encoded using ETSI-AFE, while only the singular values and the nearest left singular vectors index are encoded and transmitted for the even frames. At the server side, the non-transmitted MFCCs are evaluated through their quantized singular values and the nearest left singular vectors. The system provides a compression bit-rate of 2.7 kbps. The recognition experiments were carried out on the Aurora-2 database for clean and multi-condition training modes. The simulation results show good recognition performance without significant degradation, with respect to the ETSI-AFE encoder.

Index Terms— Distributed speech recognition, MFCC coefficients, ETSI-AFE standard, SVD decomposition

1. INTRODUCTION

The growing use of wireless and World Wide Web networks for speech communication, has led to Distributed Speech Recognition (DSR) systems being developed and standardized by the European Telecommunication Standards Institute (ETSI) [1]. As shown in Fig. 1, the fundamental idea of DSR consists of using a local client Front-End (FE) where the speech features are extracted and transmitted through a transmission network to a remote Back-End (BE) or remote server recognizer. The features used for recognition are the first 12 MFCCs $c1$-$c12$, the zeroth cepstral coefficient $c0$ and the logarithmic energy log E, in each speech frame. In the compression phase, the 14-dimentional feature vector is split into seven sub-vectors, such as: ($c1$, $c2$), ($c3$, $c4$), ($c5$, $c6$), ($c7$, $c8$), ($c9$, $c10$), ($c11$, $c12$), and ($c0$, *log* E); where each pair is quantized by Split Vector Quantization (SVQ) technique [2].

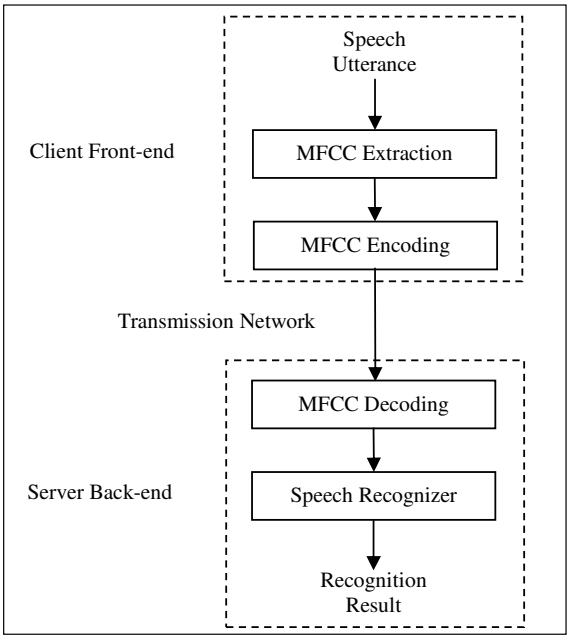

Fig. 1. MFCC encoding in DSR system.

Various schemes for compressing MFCC vectors at low bit-rate have been proposed in the literature. There are methods that investigate the inter-frame and/or the intra-frame MFCCs correlation [3-8]. In [3], the 14-dimension MFCC vectors (same coefficients as DSR standards) are grouped; where eight temporally consecutive cepstal coefficients are processed by the Discrete Cosine Transform (DCT); the achieved compression bit-rate is around 4.2

978-1-4799-2756-2/13/$31.00

kbps. The authors in [4] offer a Half Frame Rate (HFR) front-end by investigating the redundancies in the Full Frame Rate (FFR) features of ETSI front-end. The algorithm has been evaluated on the Aurora-2 clean speech; the comparisons of achieved performance accuracy levels are close to the conventional ETSI front-end compression algorithm. The work in [5] presents a scalable predictive method, in which every feature is independently quantized by providing flexibility to adjust the bit-rate according to the bandwidth requirements and server load. However, the performance is considerably reduced at low bit-rate condition.

The method proposed in [6] uses the multi-frame Gaussian Mixture Model-based block (GMM); where the method provides a performance degradation of 1% over the Aurora baseline at 0.8 kbps, for clean speech data. The multi-frame GMM-based block was extended for MFCCs compression in noisy environments [7]; although, the obtained results showed a degraded recognition performance in lower noise levels. In addition, the work in [8] proposes a packetization and variable bit-rate compression scheme, by grouping the coded MFCC frames used in the conventional DSR standard. The packetization method provides lossless compression at 3.4 kbps for clean speech data.

In this paper, we focus on reducing the compression bit-rate, by proposing a new compression scheme based on ETSI-AFE encoder and Singular Value Decomposition (SVD). The advantage of this method allows its adaptation to the existing DSR encoders in case of bandwidth reduction needs. The proposed scheme does not cause any significant performance degradation, principally in case of noisy conditions, with an acceptable computational complexity.

We exploit the slow evolution property between two successive MFCC frames factorized into SVD components, where we could transmit only the singular value and the nearest left singular vector index of the second MFCC vector. However, at the remote back-end, we have to construct each non transmitted MFCC through its de-quantized singular value and its nearest left singular vector.

The paper is organized as follow: Section 2 introduces a general overview of DSR standardizations. In Section 3, a brief review of SVD decomposition and a detailed description of the proposed SVD encoder are provided. Section 4 summarizes the experimental results. Finally in Section 5 we offer our conclusion.

2. ETSI DSR STANDARDIZATIONS

In the conventional DSR ETSI Front-End (ETSI-FE) standard [9], the 14-dimentional MFCC vectors used in the front-end part, are derived from the extracted speech frames at frame length of 25ms with frame shift of 10ms. Then, a Fourier transform is performed and followed by Mel filter bank with 23 frequency bands in the range from 64 Hz up to 4 kHz. In the compression task, the standard computes a feature vector every 10ms. As outlined in section 1, the SVQ technique is used, and it allocates 44 bits to each feature vector to achieve a total compression bit-rate of 4.4 kbps, and 4.8 kbps with including channel bit-rate.

The ETSI Advanced Front-End (ETSI-AFE) standard [10] provides considerable improvements in recognition performance, in presence of background noise. In the feature extraction part, noise reduction is performed first, which is based on Wiener filtering theory. Then, waveform processing is applied to the de-noised signal and cepstral features are calculated. Voice Activity Detection (VAD) for the non-speech frame dropping is also implemented in feature extraction. On the server side, unlike to conventional ETSI-FE standard where the delta and delta-delta coefficients are calculated via the HTK recognition engine, the ETSI-AFE includes additional scripts to compute these coefficients. The number of bits allocated to the different sub-vectors and the VAD parameter of ETSI-AFE encoder is shown in Table I.

TABLE I
BITS ALLOCATION IN ETSI-AFE ENCODER AT 4.4 KBPS

MFCC Sub-vector	Allocated bits
$c1$, $c2$	6
$c3$, $c4$	6
$c5$, $c6$	6
$c7$, $c8$	6
$c9$, $c10$	6
$c11$, $c12$	5
$c0$, $log\ E$	8
VAD	1

3. PROPOSED SVD ENCODER

The use of the proposed SVD method is motivated by, (i) the SVD energy compaction property, where the energy is compacted in the higher singular values, and (ii) the correlation property of MFCC components which allows an efficient dimensionality reduction scheme. Before describing the proposed algorithm; first, a brief review of SVD decomposition is given.

3.1. Singular Value Decomposition

SVD decomposition is an extremely powerful and useful tool in linear algebra; it is widely used in signal processing domain such as image coding and noise reduction. Let's give a matrix $\mathbf{A}$ with m rows and n columns, then there exist orthogonal matrices $\mathbf{U}$ ($m \times m$) and $\mathbf{V}$ ($n \times n$), such that:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m] \quad \text{and} \quad \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \tag{1}$$

It can be proven that [11]:

$$\mathbf{U}^{\mathrm{T}}\mathbf{A}\mathbf{V} = \mathrm{diag}(\sigma_1, \ldots, \sigma_p), \quad p = \min(m, n) \tag{2}$$

Where: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ (3)

The σ_i are the singular values of **A** and the vectors $\mathbf{u_i}$ and $\mathbf{v_i}$ are the *ith* left singular vector and the *ith* right singular vector respectively. Then **A** can be factorized into three matrices:

$$\mathbf{A} = \mathbf{USV}^{\mathrm{T}} \quad (4)$$

Here, **S** is an $m \times n$ diagonal matrix with singular values σ_i on the diagonal. If the SVD of **A** is given by (4), we define the rank r by:

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0 \quad , \mathrm{rank}(\mathbf{A}) = r \quad (5)$$

3.2. SVD Encoder

The analysis part of the algorithm is depicted by Fig 2. It can be seen that a set of feature vectors, in the speech utterance, are grouped into blocks of nine successive MFCC frames with one overlapping frame. For each block, first, the odd 14-dimentional MFCC vectors are quantized using ETSI-AFE encoder, then all the MFCCs are factorized into **U** (14×14), **S** (14×1), and **V** (1×1) matrices, such as:

$$\left\{ \begin{array}{c} \mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{14}] \\ \mathbf{V} = [\mathbf{v}_1] \quad \text{where} \quad \mathbf{v}_1 = 1 \\ \sigma_1 >= \sigma_2 = \ldots = 0 \end{array} \right\} \quad (6)$$

We should highlight that for the **U** matrix only the first column vector ($\mathbf{u}_1$) is considered, since only the first singular value is taken (σ_1). (i.e., the rank r=1).

In each MFCC block we define three sets of vectors, such as $\mathbf{U}_{Odd}$, $\mathbf{U}_{Even}$, and $\mathbf{S}_{Even}$. $\mathbf{U}_{Odd}$ contains the left singular vectors of the odd MFCC frames and their corresponding linear interpolated vectors. $\mathbf{U}_{Even}$ and $\mathbf{S}_{Even}$ contain the left singular vectors and the first singular values of the even frames respectively.

Then, $\mathbf{U}_{Odd}$ is used to select the left singular vector index of each even frame by the criterion of minimum distortion, where the Mean Square Error (MSE) is used as a distortion measure that can be expressed as:

$$\mathrm{D}_{\mathrm{MSE}}(\mathbf{u}, \mathbf{u}') = (\mathbf{u} - \mathbf{u}')^{\mathrm{T}} (\mathbf{u} - \mathbf{u}') \quad (7)$$

where, **u** and **u**' are considered as the odd left singular vector of $\mathbf{U}_{Odd}$ and the even left singular vector of $\mathbf{U}_{Even}$ respectively, and "T" is the transpose operator.

The left singular vector index of each vector in $\mathbf{U}_{Even}$ is transmitted as 3 bits, by the fact that $\mathbf{U}_{Odd}$ contains a total of eight vectors. The first singular value σ_1 of the even frames is transmitted to the back end side, since it contains more information about frame energy. The σ_1 component is encoded using uniform scalar quantization with codebook size of 256 codevectors.

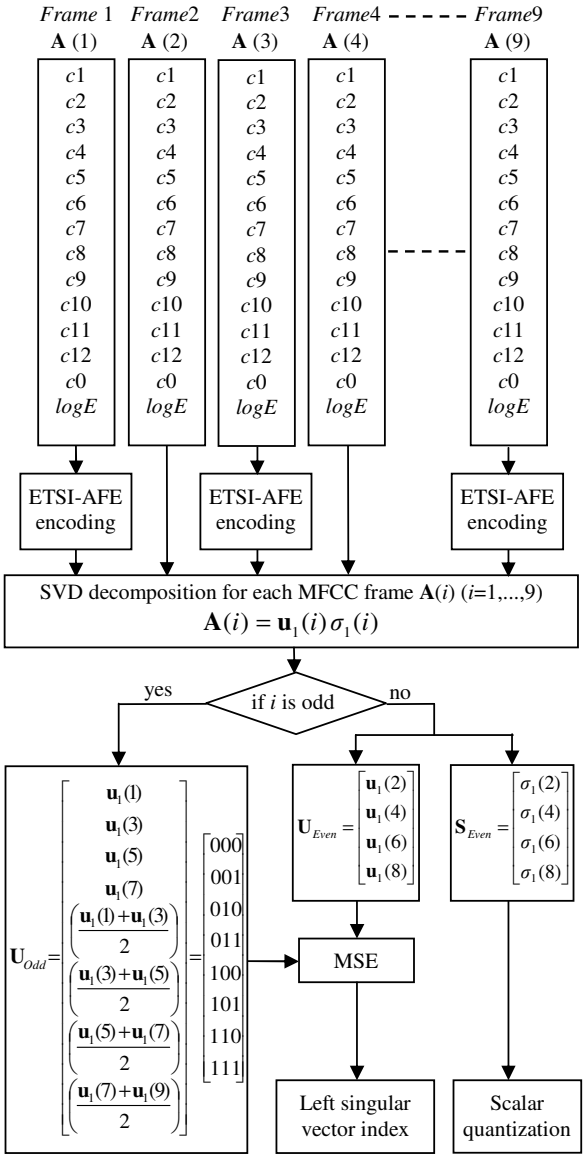

Fig. 2. SVD encoding scheme.

To perform the selection of the nearest left singular vector rather than interpolating it directly through its adjacent frames, will be more benefit in term of minimum distortion. Thus, an experiment has been performed in order to motivate the proposed scheme, where we calculated the overall distortion within blocks of successive MFCC frames extracted from the speech training utterance of Aurora-2 database [12]. As it can be shown from Fig 3, for each MFCC coefficient, the distortion measure using the proposed scheme is generally less than the case of estimating the even left singular vector directly by interpolation.

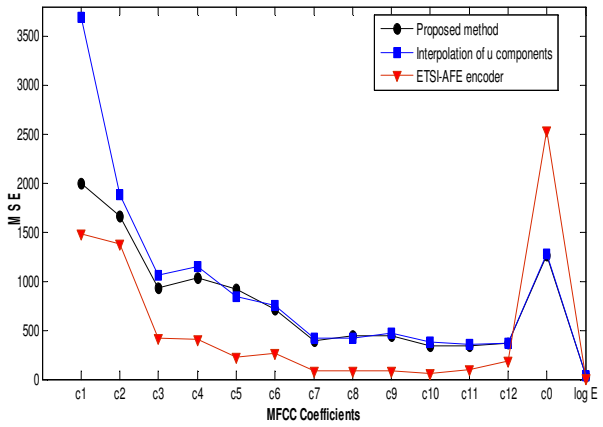

Fig. 3. MSE Distortion measure for the proposed method compared to directly interpolating left singular vectors.

In order to minimize the computational complexity, the codebook values of σ_1 component are sorted and split into four codebooks of 64 values each; hence, the scalar quantization is performed through two stages. In the first stage the nearest codebook to be used (2bits) will be selected, and the second stage for σ_1 quantization (6 bits). Table II shows the bits allocation for two successive MFCC frames with a total of 54 bits by block of 20ms. Then, the resulting quantization bit-rate is 2.7 kbps.

TABLE II
BITS ALLOCATION IN THE PROPOSED SVD ENCODER AT 2.7 KBPS, FOR TWO SUCCESSIVE MFCC FRAMES

MFCC and SVD components	Bits Allocation (odd frame)	Bits Allocation (even frame)
$c1$, $c2$	6	-
$c3$, $c4$	6	-
$c5$, $c6$	6	-
$c7$, $c8$	6	-
$c9$, $c10$	6	-
$c11$, $c12$	5	-
$c0$, *log E*	8	-
$\mathbf{u}_1$ index	-	3
σ_1	-	8
Total	43	11

4. EXPEREMENTS AND RESULTS

The MFCC are extracted by ETSI-AFE extraction algorithm, and the recognition was carried out on Aurora-2 database using HTK 3.4 speech recognition engine [13-14]. The $c0$ and *log E* coefficients are both used in the compression task and only *log E* is used in the recognition task. The results are compared for both compressed and uncompressed Aurora-2 ETSI-AFE.

Aurora-2 database provides speech samples and scripts to perform speaker independent speech recognition experiments in clean and noisy conditions. This database has been prepared by down-sampling to 8 kHz, filtering with the G.712 and MIRS characteristics. Noise is artificially added to the filtered speech utterances at a desired Signal to Noise Ratio (SNR) levels (20, 15, 10, 5, 0, - 5dB) with including clean condition, and eight different noise conditions, such as subway, babble, car, exhibition hall, restaurant, street, airport, and train station. There are three testing sets (test set A, test set B, and test set C), and two training modes, such as clean and multi-condition modes. For further details a full description of Aurora-2 database is given in [12].

The VAD parameter is not considered in the compression task, which means that the ETSI-AFE source coding bit-rate will be 4.3 kbps instead of 4.4 kbps; where the delta and delta-delta coefficients are estimated via the HTK engine. The recognition performance was measured in terms of Word Accuracy Rate (WAR), defined by:

$$WAR = \frac{N - S - D - I}{N} \times 100\ \% \qquad (8)$$

where N is the total number of words in the test set, S is the number of substitution errors, D is the number of deletion errors, and I is the number of insertion errors [14]. We should point out that test sets A and B have twice as many utterances as test set C and therefore should be given twice the weighting when calculating the WAR average [12]. Hence, the overall accuracy is calculated by:

$$WAR_{(overall)} = \frac{\left(2 \times \left(WAR_{(setA)} + WAR_{(setB)}\right) + WAR_{(setC)}\right)}{5} \qquad (9)$$

Table III and Table IV show the recognition performance average for all Aurora-2 noise conditions including clean speech (with SNR varies from 20 to -5 dB), using models trained with both compressed and uncompressed training. In the three test sets, generally the recognition performance is maintained comparing to ETSI-AFE encoder working at 4.3 kbps, with an overall degradation of 0.17% and 0.36% relative to ETSI-AFE, in case of clean and multi-condition trainings respectively (i.e., in case of compressed training.).

The computational complexity point of view shows that the proposed scheme has an acceptable computational cost since it requires half of ETSI-AFE complexity coding with additional calculations, in order to estimate the SVD components and to select the nearest left singular vector.

TABLE III
OVERALL WAR (%), IN CLEAN TRAINING MODE

Test set	SNR	ETSI-AFE baseline	ETSI-AFE Encoder (4.3 kbps)	Proposed encoder with uncompressed training (2.7 kbps)	Proposed encoder with compressed training (2.7 kbps)
Test set A	Clean	99.09	99.14	99.08	99.08
	(20-0) dB	86.70	86.33	85.90	86.00
	-5 dB	28.57	28.48	27.81	28.55
Test set B	Clean	99.09	99.14	99.08	99.08
	(20-0) dB	85.56	85.30	84.76	84.85
	-5 dB	26.77	26.29	25.31	26.05
Test set C	Clean	99.06	99.03	99.06	99.09
	(20-0) dB	82.82	82.16	81.55	82.84
	-5 dB	25.10	24.31	24.10	25.66
Overall	(20-0) dB	85.47	**85.08**	84.58	**84.91**

TABLE IV
OVERALL WAR (%), IN MULTI CONDITION TRAINING MODE

Test set	SNR	ETSI-AFE baseline	ETSI-AFE Encoder (4.3 kbps)	Proposed encoder with uncompressed training (2.7 kbps)	Proposed encoder with compressed training (2.7 kbps)
Test set A	Clean	98.97	98.96	98.90	98.90
	(20-0) dB	91.79	91.64	90.99	91.31
	-5 dB	38.92	38.37	35.93	39.61
Test set B	Clean	98.97	98.96	98.90	98.90
	(20-0) dB	90.76	90.64	90.14	90.09
	-5 dB	37.56	36.84	35.35	37.35
Test set C	Clean	98.91	98.95	98.83	98.80
	(20-0) dB	89.12	88.83	87.79	88.81
	-5 dB	30.66	30.25	28.13	31.95
Overall	(20-0) dB	90.85	**90.68**	90.01	**90.32**

5. CONCLUSION

In the proposed SVD compression scheme we focused on reducing the source coding bit-rate of MFCC vectors to 2.7 kbps. This represents around 37% of bandwidth reduction with relatively reasonable computational complexity and algorithmic delay. Generally the encoder shows a negligible degradation in term of recognition accuracy with respect to ETSI-AFE encoder working at 4.3 kbps (i.e., for compressed MFCCs).

The MFCCs correlation property has been exploited where only the odd MFCC frames are transmitted to the back end side, while the even ones are evaluated from their singular values and the estimated nearest left singular vectors. Maybe the recognition performance will be further improved if proposing new scheme for the quantization of the odd MFCCs, rather than using ETSI-AFE encoder.

Further contribution will involve improving the recognition performance and reducing the computational complexity, by proposing new criterion to select the nearest left singular vector, and a new quantization scheme for the odd frames.

6. ACKNOWLEDGMENT

The authors gratefully acknowledge the Signal Processing and Speech Communication Laboratory team for their contribution to carry out this research work.

7. REFERENCES

[1] D. Pearce, "Enabling new speech driven services for mobile devices: an overview of the ETSI standards activities for distributed speech recognition," in *Proceedings of the Voice Input/Output Applied Society Conference*, 2000.

[2] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Transactions on Acoustic, Speech, and Signal Processing,* vol. 1, no. 1, pp. 3-14, 1993.

[3] I. Kiss and P. Kapanen, "Robust feature vector compression algorithm for distributed speech recognition," in *ICSLP, International Conference on Spoken Language Processing*, 1999, pp. 2183-2186.

[4] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Adaptive multi-frame-rate scheme for distributed speech recognition based on a half frame-rate front-end, " in *MMSP*, *International Workshop on Multimedia Signal Processing*, 2005, pp. 1-4.

[5] N. Srinivasamurthy, A. Ortega, and S. Narayanan, "Efficient scalable encoding for distributed speech recognition," *Speech Communication*, vol. 48, no. 8, pp. 888-902, 2006.

[6] K.K. Paliwal and S. So, "Scalable distributed speech recognition using Gaussian mixture model-based block quantization," *Speech Communication*, vol. 48, no. 8, pp. 746-758, 2006.

[7] K.K. Paliwal and S. So, "Multi-frame GMM-based block quantization for distributed speech recognition under noisy conditions," in *ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 189-192.

[8] B.J. Borgström and A. Alwan, "A packetization and variable bit-rate interframe compression scheme for vector quantizer-based distributed speech recognition," in *Interspeech*, 2007, pp. 578-581.

[9] ETSI ES 202 050 Ver. 1.1.5, Speech Processing, "Transmission and quality aspects (STQ): distributed speech recognition; advanced fron-end feature extraction algorithm; compression algorithms," 2007.

[10] ETSI ES 201 108 Ver. 1.1.3, Speech Processing, "Transmission and quality aspects (STQ): distributed speech recognition; front-end feature extraction algorithm; compression algorithms," 2003.

[11] L.N. Trefethen and D. Bau, III, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[12] H-G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proceedings of ISCA Tutorial and Research Workshop*, 2000, pp. 181-188.

[13] "HTK Speech Recognition Toolkit," [Online]. Available: http://htk. eng. cam.ac.uk/.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK book*, HTK Ver. 3.4, Cambridge University; Engineering Department, 2006.