Waveform Speech Coding Using Multiscale Recurrent Patterns

Frederico S. Pinagé [†‡1], Lara C. R. L. Feio [†2], Eduardo A. B. da Silva [†3], Sergio L. Netto [†4]

[†] *PEE-COPPE/DEL-Poli, Federal University of Rio de Janeiro,*
POBox 68.504, Rio de Janeiro, RJ, 21945-970, Brazil.
[2] lclf@lps.ufrj.br, [3] eduardo@lps.ufrj.br, [4] sergioln@lps.ufrj.br

[‡] *Fundação Centro de Análise, Pesquisa e Inovação Tecnológica,*
Distrito Industrial, Manaus, AM, 69075-351, Brazil.
[1] frederico.pinage@fucapi.br

***Abstract*—This paper revisits the waveform paradigm for coding speech signals, using a multiscale recurrent-pattern matching approach. The so-called MMP (Multidimensional Multiscale Parser) algorithm uses a dictionary which is constantly updated with expansions, contractions, and concatenations of previously encoded segments. This provides a learning ability to the MMP, particularly suited for coding voiced and silent segments of speech. Additional features (nonuniform and auxiliary displacement dictionaries) are considered in order to adjust the MMP learning mechanism for the speech coding problem. Current MMP algorithm achieves a fair-to-good objective score when operating at 8 kbps, as indicated by several speech-coding experiments. This indicates that it may be worthy to further investigate the use of the multiscale recurrent pattern matching paradigm for speech coding.**

I. Introduction

Most speech coders/decoders (codecs) in use today are based on the CELP (code-excited linear prediction) technique [1]. Such approach employs a linear prediction model for the human speech system, which processes an input signal determined by the analysis-by-synthesis (AbS) scheme. Current CELP-based standards [2] yield toll-quality coding at a rate of about 4–10 kbps, a performance that cannot be matched by waveform codecs [3] [4]. The present paper deviates from this paradigm by performing waveform speech coding using the recently developed Multidimensional Multiscale Parser (MMP) approach [5] [6], based on multiscale recurrent pattern matching. The MMP is a multidimensional compression algorithm, that has achieved state-of-the-art result for a wide class of signals, including one-dimensional signals, as electrocardiograms (ECG) [7], two dimensional signals such as plain images [8], [10], stereoscopic images [11] and even three-dimensional signals [12]. The MMP algorithm encodes segments of a signal using a dictionary formed by expansions, contractions, and concatenations of previously encoded segments. Therefore, the MMP dictionary inherently learns the patterns present in the signal, which lends it a universal flavor. The signal is segmented and encoded according to a rate-distortion criterion. Although the MMP constitutes a waveform, time-domain codec, some criteria related to both human perception and to how human speech is produced can be incorporated in the processes of dictionary learning and segment encoding.

In this paper we analyze the coding of speech signals using an MMP-based approach. We consider a perceptually optimized encoder in the time domain, using μ-law amplitude quantization, and a displacement dictionary containing recently coded samples, which improve the coding procedure along regular portions of the speech such as voiced or silent segments. With these features, we have achieved an objective score, as determined by ITU-T Rec. P.862 PESQ (perceptual evaluation of speech quality) [9], in the range of 3.53@8 kilobits per second (kbps). This paper is organized as follows: Section II introduces the main concepts behind the MMP paradigm and Section III considers the adjustments required by the MMP approach for coding speech signals. Section IV presents several experiments illustrating the MMP quality × rate performance for a set of practical speech signals, whereas Section V closes the paper emphasizing its main contributions.

II. The MMP Algorithm

The MMP represents segments of the signal to be encoded using approximations obtained from previously encoded segments [5], [6]. This way, in the MMP algorithm there is a process of learning the patterns present in the input signal, justifying the term “recurrent patterns”. In addition, the segment matching is performed in different dimensions, considering expanded and contracted versions of the recurrent patterns. It achieves this by encoding the signal using multiple dictionaries, one for each scale. Each dictionary is updated with concatenations of dictionary words used for encoding previous segments of the signal, expanded or contracted to match each the scale of each dictionary.

In the context of speech coding, the MMP algorithm starts by dividing the one-dimensional input signal into size-N blocks. In this work, we consider $N = 128$.

Suppose we have the set of S dictionaries $\{\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(S-1)}\}$. The elements of the scale-s dictionary $D^{(s)}$ have dimension 2^s. This implies that the block size N is equal to 2^{S-1}.

In order to be encoded, each input signal block is segmented according to a segmentation tree such as the one illustrated in Fig. 1. If a tree leaf corresponds to a segment of dimensions 2^s, it is encoded using a vector $v_k^{(s)}$ from dictionary $D^{(s)}$. In this example, the block is encoded using the symbol sequence $0, 0, 1, i_3, 1, i_4, 1, i_2$, where the bits 0 and 1 represent the partition or not, respectively, of a segment. The integers i_k are the indexes, in the dictionary of scale s, of the vector $v_k^{(s)}$ used to approximate the scale s segment X_{i_k}. In this case, if the scale of X_0 is $S = 8$ (corresponding to a block size equal to $N = 2^{(8-1)} = 128$), then the input segment has

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

been approximated as $[\widehat{X}_{i_3}\widehat{X}_{i_4}\widehat{X}_{i_2}]$ by the concatenation of the codewords $v_{i_3}^{(5)}$, $v_{i_4}^{(5)}$, and $v_{i_2}^{(6)}$.

The MMP code for the segment is obtained by encoding the generated stream of symbols using a context-based adaptive arithmetic coder [13], [5]. We have different encoding contexts depending on the scale (depth level of the tree) for both flags and vector indexes. In addition, following the approach in [8], [10], the encoding of vector indexes is further conditioned by the original scale of a vector when the dictionary was updated with an expanded or contracted version of it.

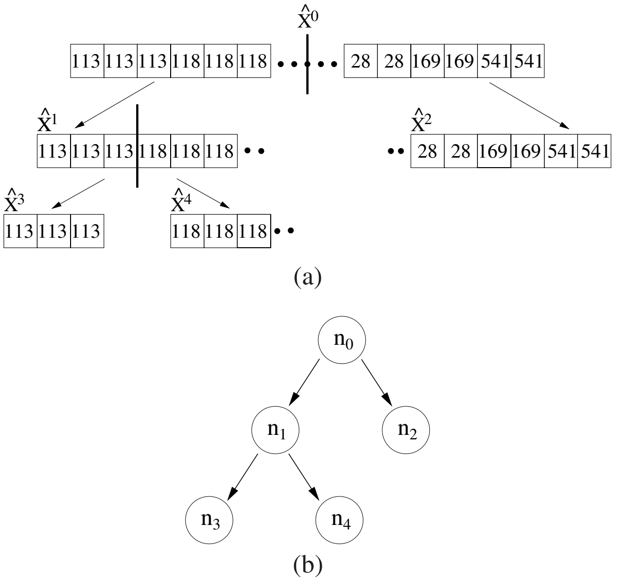


Fig. 1. (a) Example of the segmentation tree of a speech block; (b) Compact representation of the segmentation tree in (a).

After encoding, the dictionaries are updated. The procedure can be best understood by referring to Figs. 1a and 1b. As seen above, the segments corresponding to nodes n_3 and n_4 are approximated as $\widehat{X}^3 = v_{i_3}^{(5)}$ and $\widehat{X}^4 = v_{i_4}^{(5)}$, respectively. In this case, each dictionary $\mathcal{D}^{(s)}$ is updated using the approximation corresponding to their parent node n_1 ($\widehat{X}_1$ in the figure), that is the concatenation of $\widehat{X}^3$ and $\widehat{X}^4$. Since they have scale 5, that is, length $2^5 = 32$, then their concatenation $\widehat{X}_1$ has length 64, and thus a scale transformation must be carried out to transform $\widehat{X}_1$ to the scale s and include it in $\mathcal{D}^{(s)}$. For example, in order to update dictionary $D^{(4)}$, this length-64 vector must be contracted to length $2^4 = 16$. The scale transformation is a simple sampling rate change operation, and details on it can be found in [5] and [8]. In addition, all the concatenations of tree nodes with the same parent node are recursively included in the dictionaries. Again referring to Figs. 1a and 1b, the dictionaries are also updated with $\widehat{X}_0$, that is the concatenation of $\widehat{X}_1$ and $\widehat{X}_2$. However, before updating a dictionary, it is checked if there is a dictionary word too close to the one being introduced; if this is the case, the dictionary is not updated [10].

It is important to note that, since the dictionaries are updated based on the reconstructed versions of the segments, then the decoder can perform exactly the same dictionary updating operations as the encoder, and therefore the dictionary can be adapted without the transmission of any side information.

The initial dictionaries are usually very simple. We start from dictionary $\mathcal{D}^{(0)}$, that is composed of just scalars (dimension $2^0 = 1$). One way to initialize it is with all the sample amplitudes the signal may have. For example, if the dynamic range of the signal is from $A_{\min}$ to $A_{\max}$, then $\mathcal{D}^{(0)}$ will be the set $\{A_{\min}, A_{\min}+1, A_{\min}+2, \ldots, A_{\max}\}$. The initial dictionaries for the other scales are derived from $\mathcal{D}^{(0)}$ by scale transformations.

The segmentation tree is obtained using Lagrangean rate-distortion optimization [6], [14]. We minimize, for each node n_l, the Lagrangean cost $J(n_l)$ defined by

$$J(n_l) = D(n_l) + \lambda R(n_l), \tag{1}$$

where:

- $D(n_l)$ is the l_2 (squared) distortion between segment X_i corresponding to node n_l and its approximation using the dictionary of the corresponding scale;
- $R(n_l)$ is the rate spent to encode the index i_l of the vector used to approximate the segment associated to node n_l, together with the cost to represent the flags associated with node n_l;
- λ is a factor that weights the relative importance of rate and distortion in the optimization process.

In this process we compute both the cost associated to node n_l and the costs associated with its descendants n_{2l+1} and n_{2l+2}. If $J(n_l) > J(2n_l+1) + J(2n_l+2)$ the nodes n_{2l+1} and n_{2l+2} are kept and we make $J(n_l) = J(2n_l+1) + J(2n_l+2)$. Otherwise, the nodes n_{2l+1} and n_{2l+2} are pruned. Starting from the full tree, this process is recursively repeated from bottom up; when node n_0 is reached, the final segmentation tree is obtained.

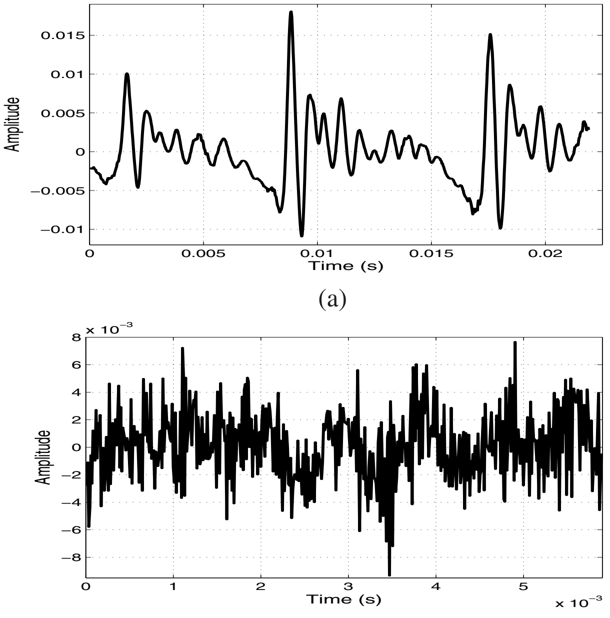
For more details on the MMP algorithm, the reader is encouraged to consult the references [5], [6], [8], [10], [15].

III. MMP-Based Speech Coding

In the field of speech coding, the MMP approach would be classified as a waveform codec, since originally it operates exclusively in the time domain [16]. The MMP efficiency is highly based on its learning ability along the dictionary updating process. In practice, however, this learning process can easily incorporate temporal, spectral, and even perceptual characteristics of any given process, such as speech, in our particular case. By doing so, the MMP algorithm ends up modeling the process in a nonparametric manner through its dictionary content.

A simple time-domain analysis of speech signals indicates that coding errors in large-amplitude samples are less perceptive than in small-amplitude intervals. For that matter, a nonuniform quantizer, following the log-like μ law [16] is employed by waveform encoders, such as the ITU-T G.711 [3], enforcing the same signal-to-quantization-noise ratio (SNR) on the entire dynamic range of the speech signal. This procedure can be incorporated into the MMP algorithm by quantizing its dictionary according to the same rule. This operation must be applied to the initial dictionary as well as in the updating stage, providing a time-domain perceptual feature to the MMP learning process. As a positive side-effect, this procedure also limits the dictionary growth, simplifying the subsequent dictionary searches and, consequently, the entire coding procedure.

A speech signal may present two levels of statistical correlation: (i) short-time correlation between consecutive samples and (ii) long-time correlation in voiced segments, that characterizes the pitch-generation process. In these voiced segments, a quasi-periodic behavior arises (see Fig. 2a), which can be easily encoded with the MMP scheme with the aid of an auxiliary displacement dictionary containing the most recently coded signal samples. If a good approximation in this displacement dictionary is found, a flag is sent to the decoder together with the delay index indicating the best segment position in this auxiliary displacement dictionary. Note that this delay index varies in one-sample intervals. With this strategy, the search procedure is greatly simplified, since just a limited set of segments are compared to the current speech signal, and the coding rate is reduced accordingly. In unvoiced intervals, as illustrated in Fig. 2b, signal tends to be quite irregular in time domain, but some regularity can still be found in the spectral or statistical domains.



(a)

(b)

Fig. 2. Basic examples of speech segments: (a) voiced; (b) unvoiced.

IV. Experimental Results

In this section, we investigate the MMP rate-distortion performance when coding speech signals. In all experiments shown here, we consider database DB1 consisting of 10 sentences, phonetically balanced to the Brazilian Portuguese language [17], sampled at 8 kHz and with a 16-bit precision. Resulting quality of the MMP codec is assessed objectively using ITU-T PESQ measure [9] mapped onto the mean opinion score (MOS) 1–5 scale.

Experiment 1: In a first experiment, we investigate how the μ-law quantization on the MMP dictionary influences the algorithm performance when coding database DB1. Fig. 3 shows the PESQ-MOS results when using the MMP algorithm with uniform or μ-law dictionaries. From this figure, we observe how the nonuniform dictionaries greatly improve the MMP rate-distortion trade-off as compared to the uniform dictionary. As mentioned above, computational complexity is greatly reduced by the nonuniform dictionaries, since each of the different scales dictionaries are initialized with 257 elements (256 levels plus the null amplitude). On the other hand, the uniform dictionaries for each scale start with 4096 elements. As far as the dictionary growth is concerned, for the 8-kbps rate, the uniform dictionary increases to around 85000 elements, after encoding each sentence whereas its nonuniform counterpart grows only to 53000, indicating that the nonuniform search is more effective throughout both learning processes.

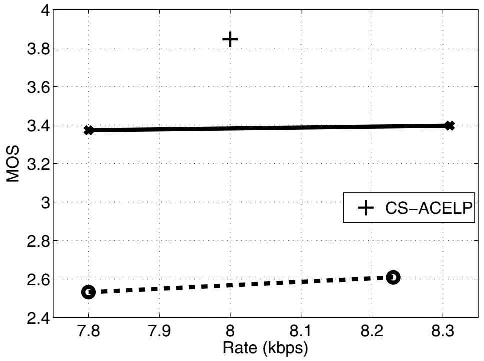


Fig. 3. PESQ-MOS results for the MMP algorithm, with coding rate around 8 kbps, with uniform (dashed line) and nonuniform (solid line) dictionaries, in comparison to G.729 results.

Experiment 2: Fig. 4 shows the results of the MMP algorithm incorporating the auxiliary displacement dictionary, for different values of its length L. It is verified how this feature improves the overall MMP performance, particularly for $L = 1024$, which achieves the best quality × rate compromise in the 8-kbps range.

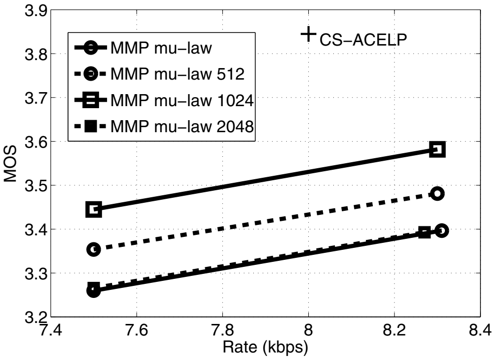


Fig. 4. PESQ-MOS results for μ-law MMP algorithm, with coding rate around 8 kbps, for different lengths L of the displacement dictionary, in comparison to G.729 results.

The activation process of the auxiliary displacement dictionary in the MMP coding procedure is illustrated in Fig. 5. The clear areas in the lower plot of this figure indicate how the auxiliary displacement dictionary is more commonly employed in the speech segments that match previously coded samples. In this way, a highly regular signal results in larger segments being matched by the displacement dictionary, thus reducing the resulting coding rate.

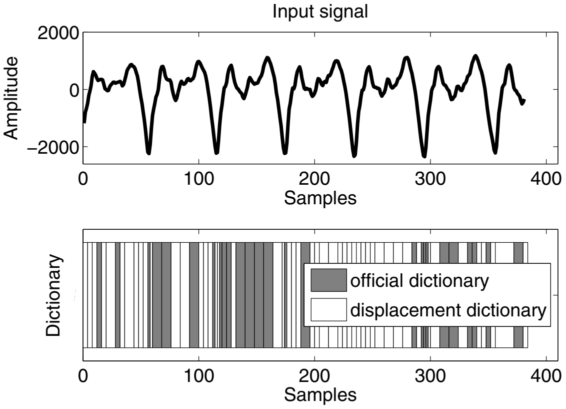


Fig. 5. Use of displacement dictionary along time in MMP coding for a given speech signal.

For the entire database DB1, the original multiscale dictionaries got activated in 70% of all segments, while the auxiliary displacement dictionary got activated in 30% of them, reducing the overall coding rate accordingly.

Fig. 6 presents the statistical distribution of the encoded segment length for the same database. From it, one concludes that all segmentation levels are used with equivalent frequencies, thus reforcing the usefulness of the multiscale nature of the dictionaries in speech coding.

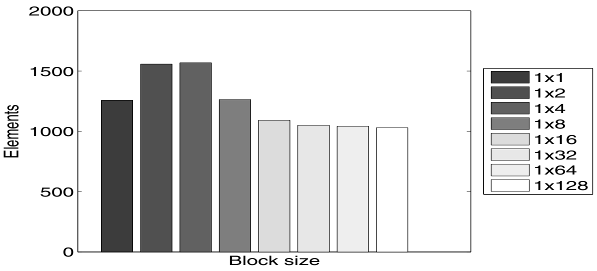


Fig. 6. Statistics of segment size in MMP coding for speech database DB1.

Experiment 3: The overall performance of the MMP algorithm for different coding rates of database DB1 is depicted in Fig. 7. For the sake of comparison, results are also shown for the ITU-T G.711 (PCM) [3], ITU-T G.726 (ADPCM) [4] and ITU-T G.729 (CS-ACELP) [2] speech codecs. The MMP version with non-uniform initial dictionary (solid line) approximates the G.729 performance. Note that the version with initial 12 bits uniform dictinary achieves a signal to noise ration slightly above the one of PCM, but at a higher rate.

V. Conclusion

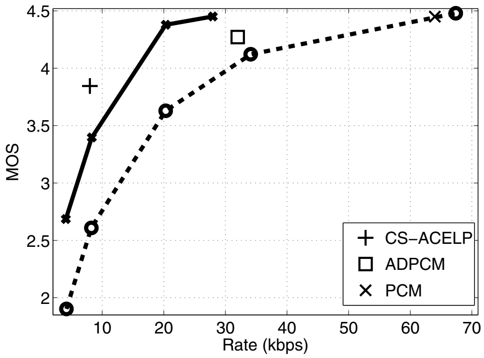


Fig. 7. PESQ-MOS × coding rate for the MMP algorithm.

We investigate the performance of the multiscale recurrent pattern matching (MMP) approach for encoding speech signals. Two features (nonuniform and auxiliary displacement dictionaries) were incorporated to the standard MMP method, increasing its efficiency in terms of quality and rate for the particular problem at hand. With such improvements, current MMP codec achieves a PESQ-MOS result of 3.53 when operating at 8 kbps. Additional features, such as voiced/unvoiced segment classification and perceptual (PESQ-like) segmental search, can be easily incorporated into the MMP algorithm, indicating that this research line may be worth pursuing as a viable alternative for modern codecs.

References

[1] M. Schroeder and B. Atal, “Code excited linear prediction: High quality speech at low bit rates,” *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing,* pp. 937–940, Tampa, USA, 1985.

[2] ITU-T Rec. G.729, *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, 1996.

[3] ITU-T Rec. G.711, *Pulse Code Modulation (PCM) of Voice Frequencies*, 1983.

[4] ITU-T Rec. G.726, *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation*, 1990.

[5] M. B. de Carvalho, “*Multidimensional Signal Compression using Multiscale Recurrent Patters*,” Ph.D. Thesis, COPPE/UFRJ, Mar. 2001.

[6] M. B. de Carvalho, E. A. B. da Silva, and W. A. Finamore, “Multidimensional signal compression using multiscale recurrent patterns,” *Special Edition in Image Coding Beyond Standards, Elsevier*, vol. 82, no. 11, pp. 1559–1580, Nov. 2002.

[7] E. B. de L. Filho, E. A. B. da Silva, M. B. de Carvalho, W. S. da Silva, Jr., and J. Koiller, “Eletrocardiographic signal compression using multiscale recurrent patterns,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 12, pp. 2739–2753, Dec. 2005.

[8] E. B. de L. Filho, E. A. B. da Silva, M. B. de Carvalho, and F. S. Pinagï£¡, “Universal image compression using multiscale recurrent patterns,” *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 512–527, Apr. 2008.

[9] ITU-T Rec. P.862, *Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs*, 2001.

[10] N. M. M. Rodrigues, E. A. B. da Silva, M. B. Carvalho, S. M. M. de Faria, and V. M. M. da Silva, “On dictionary adaptation for recurrent pattern image coding,” *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1640–1653, Sept. 2008.

[11] M. H. V. Duarte, M. B. de Carvalho, E. A. B. da Silva, C. L. Pagliari, and G. V. Mendonï£¡a, “Multiscale recurrent patterns applied to stereo image coding”, *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, no. 11, pp. 1434-1447, Nov. 2005.

[12] A. Frauche, M. B. de Carvalho, and E. A. B. da Silva, “3D weather radar image compression using multiscale recurrent patterns”, *2008 IEEE International Conference on Image Processing*, pp. 1049–1052, San Diego, USA, Oct. 2008.

[13] I. H. Witten, R. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the Association for Computing Machinery*, vol. 30, no. 6, pp. 520–540, June 1987.

[14] A. Orgeta and K. Ramchandran, “Rate-distortion methods for image and video compression”, *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[15] http://www.lps.ufrj.br/profs/eduardo/MMP/.

[16] J. R. Deller, Jr., J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, IEEE Press, Piscataway, USA, 2000.

[17] A. Alcaim, J. A. Solewicz, and J. A. de Moraes, “Phone occurrence rates and lists of phonetically balanced sentences for Brazilian Portuguese spoken in Rio de Janeiro,” (in Portuguese), *Revista da Sociedade Brasileira de Telecomunicações*, vol. 7, no. 1, pp. 23–41, Dec. 1992.