Video Compression Using 3D Multiscale Recurrent Patterns

Nelson C. Francisco[1,3], Nuno M. M. Rodrigues[1,2], Eduardo A. B. da Silva[3],
Murilo B. de Carvalho[4], Sérgio M. M. de Faria[1,2].

[1]Instituto de Telecomunicações, Portugal; [2]ESTG, Instituto Politécnico Leiria, Portugal;
[3]PEE/COPPE/DEL/Poli, Univ. Fed. Rio de Janeiro, Brazil; [4]TET/CTC, Univ. Fed. Fluminense, Brazil.
e-mails: nelson.carreira@co.it.pt, nuno.rodrigues@co.it.pt, eduardo@lps.ufrj.br,
murilo@telecom.uff.br, sergio.faria@co.it.pt.

***Abstract*—In this paper, we propose a new 3D pattern matching based video compression algorithm. Spatiotemporal prediction tools are used to exploit both the temporal and the spatial redundancies, and the resulting residue is encoded using a 3D data coding extension of the Multidimensional Multiscale Parser (MMP) algorithm. MMP was originally proposed as a generic lossy data compression algorithm. A high degree of adaptivity and versatility allowed it to be competitive with state-of-the-art transform-based compression methods for a wide range of applications and data sets.**

The performance of the proposed algorithm is compared with that of H.264/AVC, showing competitive results for some data sets, which indicates the potential of the pattern matching paradigm as an alternative to the traditional hybrid video codecs.

***Index Terms*—Video Coding, Pattern Matching.**

I. Introduction

A video sequence is a temporal succession of image frames, and can thus be conceived as a 3D signal, with one temporal and two spatial dimensions. Generally, a high degree of spatial and temporal correlation exists, and the success in exploiting such redundancies is the key feature for the rate-distortion performance of a video codec.

Most modern video codecs, such as H.264/AVC [1], rely on the hybrid architecture, using frame by frame motion compensation to exploit the temporal correlation, and some two-dimensional compression method to encode the resulting residue and extract the remaining spatial redundancy. However, despite the high compression efficiency achieved by some hybrid codecs, motion compensation presents some impairments for certain applications. It is a very computationally demanding operation, and it does not perform well for some types of movements, such as non-translational motion, which includes rotations, zoom and shearing of objects. This motivated the research for alternative approaches to efficiently exploit the spatiotemporal redundancy.

Several works suggested to approach video signals from a volumetric point-of-view, using straightforward 3D extensions of well-known 2D compression methods. This corresponds to process the video data directly as a 3D signal, instead of using a frame-by-frame approach. For example, the use of a 3D fractal for video compression was proposed in [2], and several researchers suggested to use 3D extensions of the Discrete Cosine Transform (DCT) [3], [4] or Discrete Wavelet Transform (DWT) [5], [6] for video compression.

In earlier proposals, 3D transforms were applied directly to the input video data [3]. Despite its efficient representation for slow movements, where the energy concentrates on the low frequency temporal coefficients, the performance of such methods degrades considerably in the presence of complex and non-uniform motion, where the energy spreads along the higher frequency temporal coefficients. This motivated the study of an alternative class of algorithms, which perform some kind of motion compensation before applying the transform [4], [5]. However, despite the good complexity *vs.* compression performance ratio achieved by some of these algorithms, none of them resulted in a competitive alternative to the state-of-the-art hybrid codecs.

In this paper, we propose a new video compression framework, referred to as 3D-MMP, which combines spatiotemporal prediction techniques with a 3D extension of the Multidimensional Multiscale Parser (MMP) algorithm, to encode the resulting residue. MMP was already used to replace transforms on a hybrid video codec [7], with results that outperform ones of the state-of-the-art H.264/AVC, demonstrating the potential from such approach. 3D-MMP combines volumetric video coding and pattern matching, giving rise to a framework competitive with H.264/AVC. In Section II we describe the proposed 3D architecture, briefly introducing the MMP algorithm and the prediction employed, while in Section III we present the experimental results. The conslucions are given in Section IV.

II. Proposed Compression Architecture

Considering k_1 and k_2 the spatial coordinates, where $1 \leq k_1 \leq H$ and $1 \leq k_2 \leq W$, and the temporal coordinate t as a generic k_3, a video sequence can be conceived as a 3D volumetric signal $X(k_1, k_2, k_3)$. Thus, groups of N fames can be encoded in a raster scan order using 3D blocks.

However, similarly to hybrid video codecs, the proposed algorithm uses a hierarchical compression approach. Instead of

This project was funded by FCT - "Fundação para a Ciência e Tecnologia", Portugal, under the grant SFRH/BD/45460/2008, and Project COMUVI (PTDC/EEA-TEL/099387/2008).

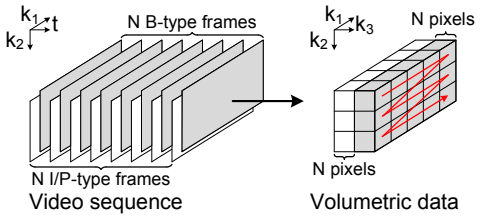


Fig. 1. Hierarchical codec architecture.

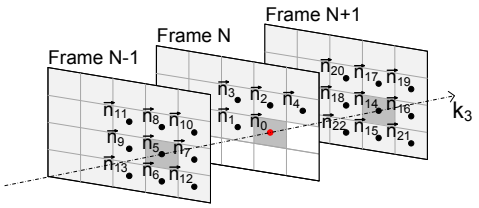


Fig. 2. Spatiotemporal neighborhood for B-type frame pixels.

sequentially processing groups of N successive frames, alternate frames are used to define each group. Fig. 1 illustrates the case where the first group of frames comprises the even frames of the video sequence, while the second group comprises the odd frames. Thus, frames from the first group can be used as reference to predict the frames in the second group, temporally located between them. This situation is similar to the I/P and B slices on H.264/AVC [1]. Thus, we refer to the frames from the first group as I/P-type frames, and those from the second group as B-type-frames.

A strong correlation can be established between each pair of N P-type and B-type frames and the GOP size in H.264/AVC. Both are the minimum temporal unit which can be decoded independently, providing random access to the video sequence, which is essential for most practical applications.

A. The MMP algorithm

MMP [8] is based on approximations of data blocks X^l of scale l, using codewords from an adaptive dictionary $\mathcal{D}^l$. For each input block, the dictionary codeword S_i^l that better represents X^l is chosen, based on an Rate-Distortion function J, given by $J = D + \lambda R$, where λ is a *Lagrangian multiplier* [9], that weights the rate R of the representation, over its resulting distortion D.

The input block is then segmented into two sub-blocks of a lower scale, X_1^{l-1} and X_2^{l-1}, each with half the pixels, and the two halves are recursively optimized. The sum of the resulting costs is then compared with that of the original block, in order to decide whether to segment or not X^l. The same procedure is recursively applied down to 1×1 sub-blocks.

The 3D layout allows to generically segment each block along the three different coordinates, provided that each half has the same number of pixels. All the possible segmentation options are tested for each block, selecting the one with the lowest Lagrangean cost, and the resulting segmentation pattern is represented by a binary tree $\mathcal{T}$. Each non-segmented block corresponds to a tree leaf, while each segmentation creates two child nodes, n_i^l. Once $\mathcal{T}$ is optimized, it is converted into a string of symbols, using a top-down approach. All the generated symbols are encoded using an adaptive arithmetic encoder, with separate contexts for each symbol type and scale.

Every segmentation of a block of scale l originates a new pattern, formed by the concatenation of two codewords of scale $l-1$. The use of a separable scale transformation T_l^s allows to adjust the vector's dimensions and update the new pattern in all of the dictionary's scales, resulting in an *adaptive dictionary* that does not require any extra overhead to be transmitted. The decoder is able to keep a synchronized copy using only the bitstream information.

In order to further improve the efficiency of the adaptive dictionary, several design techniques were proposed for 2D coding in [8], and adapted for the 3D layout. Additional codewords become available for each extra pattern created, using *geometric transforms* of the original block. At the same time, redundancy control tools are used in order to avoid the insertion of useless blocks in the dictionary [8].

B. Spatiotemporal prediction

The proposed method combines MMP with 3D hierarchical prediction. For each input block, a prediction is applied and the MMP optimal representation of the resulting residue is determined. The block is then segmented and the same procedure is applied to each half, down to a pre-established minimum prediction scale. As the choice of the best prediction and residue coding combination is performed based on the Lagrangean cost of the representation, the algorithm is able to achieve the optimal representation for each block.

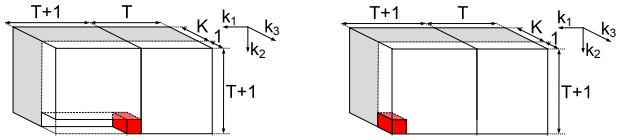
Several spatiotemporal prediction modes have been developed. One of these modes generates a least squares prediction for each pixel, on the behavior of its spatial and temporal neighbors, as proposed in [10]. Some modifications of the original method were proposed in [11], in order to adapt the pixel-by-pixel approach for two-dimensional block-based architectures. Predicted values are used instead of the reconstructed values for the pixels belonging to the same block as the one being predicted, and the training region is adapted on a block-by-block basis so that it remains in the block's causal neighborhood. These modifications were further extended for 3D blocks in the proposed method.

The prediction for a given pixel $X(\vec{n}_0)$ located at positions $\vec{n}_0$, is obtained through the equation:

$$\hat{X}(\vec{n}_0) = \sum_{i=1}^{N} a_i X(\vec{n}_i), \tag{1}$$

where $\vec{n}_i$, $i = 1, ..., N$ are a set of spatiotemporal causal neighbors. We adopted the same neighborhood proposed in [10], with some modifications, as illustrated in Fig. 2. For I/P-type frames, 4 spatial and 9 temporal neighbor pixels from the previous frame are used. For B-type frames, 9 additional neighbor pixels from the future frame are also included in the filter's support, allowing to generate a weighted implicit bi-prediction for those pixels.

Note that for pixels near the block boundaries, the neighbors on their right or bottom may be unavailable. In those cases,



(a) (b)

Fig. 3. Spatiotemporal training region (a) standard (b) rightmost column.

the pixels used on the support are displaced to the top or left, respectively.

Assuming the Markov property on a spatiotemporal causal neighborhood, the optimal prediction coefficient vector $\vec{a} = [a_1, ..., a_n]^T$ is trained to minimize the prediction error in this neighborhood, presented in Fig. 3, where $T = 5$ and $K = 3$. This training region is a volumetric extension of the one proposed in [11]. All the samples from the training region are then placed into an $M \times 1$ column vector $\vec{y}$, with M being the number of pixels on the training region. If we put the N causal neighbors for each training sample into a $1 \times N$ row vector, then all training samples generate a data matrix C, of size $M \times N$. The optimal prediction coefficients $\vec{a}$ are then determined by solving the following least squares problem:

$$\min(||\vec{y} - C\vec{a}||^2), \tag{2}$$

which has the well-known closed form solution:

$$\vec{a} = (C^T C)^{-1}(C^T \vec{y}). \tag{3}$$

A 3D directional prediction mode was also included in the proposed algorithm. Considering a generic 3D block $X^l(k_1, k_2, k_3)$, and assuming the reconstructed frame from the temporal position immediately before the frame being encoded is always available, a directional prediction can be generated for $X^l(k_1, k_2, k_3)$ using the expression:

$$\hat{X}^l(k_1, k_2, k_3) = \hat{X}^l(k_1 - v_1, k_2 - v_2, k_3 - 1), \tag{4}$$

where v_1 and v_2 are the components of a two-dimensional vector, respectively along k_1 and k_2. In other words, if we consider that the block $X^l(k_1, k_2, k_3)$ is sliced along the k_3 axis, each slice of the prediction block will assume the values of the same coordinates of the previous frame, displaced by $\vec{v}$. Note that motion estimation is itself a particular case of this approach, where the size of the block along k_3 is one and a separate vector is required for each slice.

However, the block-based approach has some inherent limitations due to causality. As each block may comprise several frames which are encoded together, the displacement in Eq. 4 may lead to the situation that portions of a frame may belong to blocks that were not encoded yet. This problem is solved by using references from the closest available frame, instead of the frame immediately before the one being predicted, that is $(k_3 - 1)$ in Eq. 4 may be replaced by $(k_3 - p)$, $p > 1$.

The choice of the optimal directional vector (DV) is performed based on the minimization of the residue's energy. The DVs use integer precision, with each component being restricted to the interval $[-4; 4]$. Alternatively, the components of the DVs can also be estimated, using the block's spatiotemporal neighborhood. Based on the Lagrangean cost of the reconstruction, the algorithm chooses between using the estimated DV, signaled by the flag 0, or by transmitting the optimal DV, signaled by the flag 1 followed by its components, encoded using an adaptive arithmetic coder.

Additionally to the previously described prediction modes, 3D extensions of the Intra modes used on H.264/AVC [1] were also adopted. For that purpose, the block is sliced along k_3 and a prediction is generated for each slice, using the H.264/AVC [1] Intra prediction modes.

III. Experimental Results

In order to assess the performance of the proposed video compression algorithm, this section presents a comparison with JM17.1 H.264/AVC reference software.

A set of commonly used parameters was adopted for H.264/AVC [1], namely a GOP size of 15 frames with an *IBBPBBP* pattern, at a frame-rate of 30fps. The high profile was defined, with the RD optimization and Intra MB in inter-predicted frames being enabled, as well as the context-based adaptive arithmetic coder (CABAC). The error resilience tools and the weighted prediction were disabled. The ME was performed using the Fast Full Search algorithm, with ± 16 search range and 5 reference frames. The variable bit rate mode was used, with the QP parameter being set separately for the I/P and B slices. Four combinations of QP values were used, namely 23-25, 28-30, 33-35 and 38-40.

For the proposed algorithm, we adopted $8 \times 8 \times 8$ blocks, in order to restrict the computational complexity, with a maximum dictionary size of 5000 elements per scale. The hierarchical frame architecture was adopted, interleaving one B-type and one I/P-type frame, sequentially. Such as for H.264/AVC, spending more bits while encoding the I/P slices revealed beneficial, as it results in better predictions for B slices. Thus, the λ used while coding the B-type blocks is set to be 50% larger than the used for the I/P-type blocks. Four combinations of λ's were used, in order to obtain rate-distortion points in the same bitrate range than that resulting from H.264/AVC, namely 20-30, 75-112, 200-300 and 500-750, respectively for the I/P and B-type blocks. The deblocking filter proposed in [12] is applied after encoding each group of eight frames.

The results are summarized in Table I. As can be seen, the proposed method is able to outperform the state-of-the-art H.264/AVC for the sequence Container. This sequence presents a uniform translational motion, which is efficiently predicted using the proposed directional prediction mode. The algorithm is able to predict several frames with the same DV, achieving a more efficient representation than the traditional ME. For example, for $\lambda = 200$, the directional prediction mode is used for 99.6% of the pixels from the B-type frames, and the high correlation between the best DV for each block and those from its neighbors results in their efficient prediction. This contributes to reduce their average entropy, which is in average, 0.33 and 0.15 bits for the k_1 and k_2 components, respectively. As a result, the rate required to encode the B-type blocks corresponds to only 10% of the overall bitrate, demonstrating the efficiency of the hierarchical architecture for

TABLE I
COMPARISON OF THE GLOBAL R-D PERFORMANCES OF 3D-MMP AND H.264/AVC JM 17.1. THE BD-PSNR CORRESPONDS TO THE PERFORMANCE GAINS OF 3D-MMP OVER H.264/AVC.

		H.264/AVC				3D-MMP				BD-PSNR		
	QP [I/P-B]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**Y** [dB]	**U** [dB]	**V** [dB]
Akiyo	23-25	256.12	43.39	45.46	46.68	272.66	42.95	46.01	47.22	**-0.91**	**0.27**	**0.42**
	28-30	140.84	40.64	42.69	44.12	144.35	39.81	43.16	44.77			
	33-35	81.06	37.65	40.07	41.82	98.56	37.59	41.28	43.00			
	38-40	48.22	34.47	38.29	40.47	58.69	35.23	38.71	40.97			
Coastguard	23-25	2335.07	38.78	45.79	46.88	2220.32	36.09	45.13	46.26	**-2.03**	**-0.17**	**-0.14**
	28-30	987.19	34.19	44.16	45.10	969.46	31.97	43.72	44.65			
	33-35	431.83	31.11	42.59	43.49	507.94	29.54	42.80	43.76			
	38-40	172.47	28.34	40.50	41.31	208.84	27.66	41.61	42.50			
Container	23-25	576.35	40.38	45.00	45.09	505.28	39.92	45.65	45.73	**0.17**	**1.05**	**0.96**
	28-30	286.29	37.02	42.26	42.31	247.71	36.58	42.85	42.90			
	33-35	146.99	33.99	39.82	39.79	145.82	34.08	40.89	40.57			
	38-40	76.99	30.94	38.32	37.98	85.78	31.52	38.97	38.58			

video compression. Note that in this case, H.264/AVC needs to transmit a vector corresponding to each block for each frame, resulting in a less efficient representation.

For the sequence Akiyo, the almost static background is also efficiently predicted by the proposed method, with 99.8% of the B-type blocks being encoded using the directional prediction mode. The average entropy for the DVs are respectively 0.49 and 0.89 bits, for the k_1 and k_2 components. However, H.264/AVC is also very efficient while encoding this static background, as it mostly uses the copy and skip modes, which require the transmission of very little information. Thus, H.264/AVC is in this case able to achieve a performance advantage of 0.9dB.

The fast panning and the existence of several moving objects in the sequence Coastguard, accompanied by some camera jittering, means that the search for adequate matches for multiple frames is more difficult. Thus, the algorithm needs to segment the block along the temporal axis, converging to the traditional frame-by-frame ME. However, for a frame-by-frame estimation, the proposed algorithm is not able to achieve the same prediction quality obtained by the more complex quarter pixels ME used in H.264/AVC, which also benefits from larger searching windows. Furthermore, the high detail present in this particular sequence imposes the need of performing more segmentations, resulting on average in smaller blocks and consequently, in a larger number of DVs to be transmitted. These DVs are also more difficult to predict, due to the erratic motion, increasing the average entropy of the k_1 and the k_2 components to 1.78 and 0.51 bits, respectively.

IV. CONCLUSIONS

In this paper, we present a new MMP-based video compression framework. The proposed algorithm adopted a hierarchical volumetric prediction, with the 3D resulting residue being encoded using a volumetric extension of the MMP algorithm.

The proposed framework was compared with H.264/AVC, presenting a competitive coding performance in most cases. This demonstrates the potential of the proposed approach, which deserves further investigations.

H.264/AVC still outperforms the proposed method for video sequences presenting fast complex motion, but we believe that the development of more sophisticated directional prediction modes may improve the performance of the proposed algorithm. Furthermore, the development of more efficient compression schemes for the vectors can contribute to reduce the rate associated to their transmission, which in some cases corresponds to almost half of the total bitrate. A multi-level hierarchical bi-prediction may also be investigated in the future.

REFERENCES

[1] ITU-T, ISO/IEC JTC 1, *Advanced video coding for generic audio-visual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 2: Jan. 2004, Version 3: Sept 2004, Version 4: July 2005.*

[2] A. Chabarchine and R. Creutzburg, "3D fractal compression for real-time video," in *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, 2001, pp. 570–573.

[3] T. Fryza, "Compression of video signals by 3D-DCT transform," *Institute of Radio Electronics, FEKT Brno, University of Technology, Czech Republic*, 2002.

[4] N. Bozinovic and J. Konrad, "Motion analysis in 3D DCT domain and its application to video coding," *Signal Processing: Image Communication*, vol. 20, pp. 510–528, July 2005.

[5] S.-J. Choi and J. Woods, "Motion-compensated 3-D subband coding of video," *Image Processing, IEEE Transactions on*, vol. 8, no. 2, pp. 155 –167, feb 1999.

[6] B.-J. Kim, Z. Xiong, and W. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 8, pp. 1374 –1387, dec 2000.

[7] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, and S. de Faria, "Efficient recurrent pattern matching video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 8, pp. 1161–1173, aug. 2012.

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

[9] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, pp. 23–50, November 1998.

[10] X. Li, "Least-square prediction for backward adaptive video coding," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 1, pp. 126–126, 2006.

[11] D. Graziosi, N. Rodrigues, E. da Silva, S. de Faria, and M. de Carvalho, "Improving multiscale recurrent pattern image coding with least-squares prediction," in *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, November 2009.

[12] N. Francisco, N. Rodrigues, E. da Silva, and S. de Faria, "A generic post-deblocking filter for block based image compression algorithms," *Signal Processing: Image Communication*, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596512001087