To be published in PCS 2004, San Francisco, CA, December 2004.

Scalable Motion-Adaptive Video Coding with Redundant Representations

Adel Rahmoune, Pierre Vandergheynst and Pascal Frossard
Swiss Federal Institute of Technology EPFL
Signal Processing Institute
CH- 1015 Lausanne, Switzerland
{adel.rahmoune,pierre.vandergheynst,pascal.frossard}@epfl.ch

***Abstract*— This paper presents a scalable video coding scheme (MP3D), based on the use of a redundant 3-D spatio-temporal dictionary of functions. The spatial component of the dictionary consists of directional and anisotropically scaled functions, which form a rich collection of visual primitives. The temporal component is tuned to capture most of the energy along motion trajectories in the video sequences. The MP3D video coding first finds motion trajectories. It then applies a spatio-temporal decomposition using an adaptive approximation algorithm based on Matching Pursuit (MP). The coefficients and the function parameters are quantized and coded in a progressive fashion, under multiple rate constraints, allowing for adaptive decoding by simple bit-stream truncation. The motion fields are losslessly coded and transmitted as side information to the decoder. The multi-resolution structure of the dictionary allows for flexible spatial and temporal resolution adaptation. This scheme is shown to yield comparable rate-distortion performances to state-of-the-art schemes, like H.264 and MPEG-4. It represents a promising alternative for low and medium rate applications, or as a flexible base layer for higher rate video systems.**

I. Introduction

Flexible representations that generate scalable video coding schemes are nowadays getting quite a lot of attention from the research community. They provide interesting solutions to an increasing number of applications that require adaptive signal representations, like video delivery over heterogeneous networks such as Internet. Successful scalable video coding schemes are generally based on the 3-D wavelet transform, and employ a separable 2-D wavelet transform (DWT) for the spatial information, and DWT with either a transversal or lifting implementation along motion trajectories [1]–[3]. Recently however, it was pointed out that the separable 2-D wavelet transform is not ideally suited for representing images as it fails to capture regular geometric features (e.g. edges) [4], mainly because it lacks directionality and anisotropic scaling. Moreover, it is a shift-variant transform, which is not desirable for representing motion in the video signal.

Three-dimensional Matching Pursuit video coding has recently been introduced in [5], as an alternative to wavelet-based scalable video coding methods. Redundant expansions with dictionaries adapted to natural image features allow for efficient coding at low bit rate. In the same time, the Matching Pursuit algorithm [6] provides high flexibility in the signal representation, in addition to an inherent progressiveness of the bit-stream structure. This paper proposes an enhanced video coding scheme that adds motion estimation to the 3D Matching Pursuit algorithm, and thus clearly improves the compression performance, especially for high motion sequences. The Motion-Adaptive 3-D Matching Pursuit algorithm advantageously uses a spatio-temporal transform where spatial atoms follow motion trajectories. At low and medium bit rates, (i.e. less than 500 kbps), the compression results are comparable to the state-of-the-art coders like H.264 and MPEG-4, in terms of rate distortion performance and visual quality.

This paper is organized as follows. The motion-adaptive 3-D transform and the embedded coding are presented in Section II. Section III presents coding experiments, carried out on standard test sequences. Section IV highlights the resolution and SNR scalability properties. Finally, conclusions and discussions are given in Section V.

This work has been supported by the Swiss National Science Foundation.

II. The MP3D Coding Scheme

A. Overview

The building blocks of the motion-adaptive three-dimensional Matching Pursuit video encoder (MP3D) are represented in Figure 1. It basically consists in two main modules, which are:

- The motion-adaptive 3-D spatio-temporal transform,
- The embedded quantization and coding.

The video sequence is first partitioned into groups of pictures (GOP) of size N_{GOP} (with $N_{GOP} = 16$ in this work). A motion estimation is performed in the GOP, in order to define the motion fields for each frame, and generate motion trajectories along the successive frames. A Matching Pursuit algorithm then determines the most relevant components of the spatio-temporal video signal. It provides a sparse representation of the video information in a series of spatial atoms that are filtered and displaced along the motion trajectories. In a sense, this operation is similar to the the motion-compensated temporal filtering (MCTF) [1], where the signal is filtered in the temporal dimension along a given trajectory. Finally, the atom parameters are then quantized and progressively encoded to generate a scalable video stream. Lossless coding (DPCM and arithmetic coding) is applied to the motion field parameters, that are sent as a constant rate side information layer to the decoder [7].

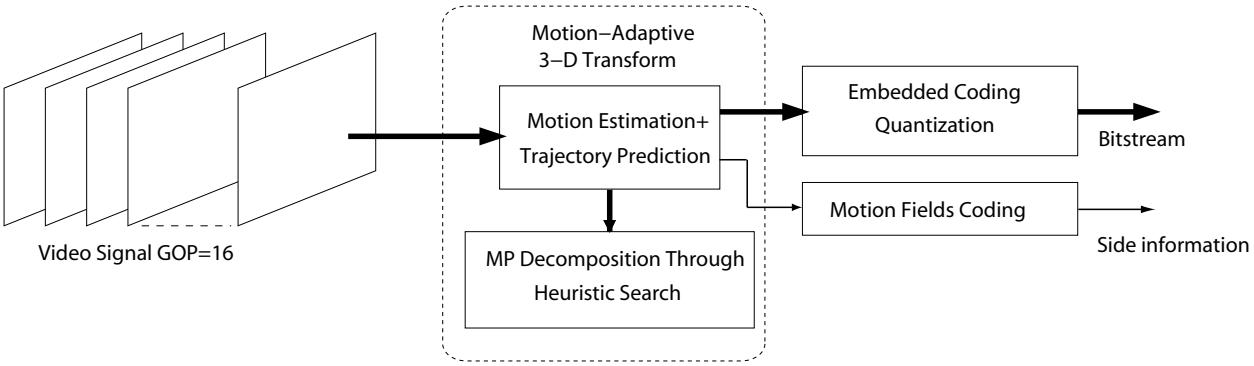

Fig. 1. Block diagram of the Motion-Adaptive Matching Pursuit encoder (MP3D).

B. Motion Estimation and Trajectory Prediction

The motion vectors are estimated using a block-matching (BM) based technique, with an exhaustive search algorithm. The motion mappings and atom trajectories are then built from the generated motion fields. A motion trajectory is defined as the displacement of a pixel or a group of pixels in a frame, toward previous and successive frames. It can be determined uniquely given both the forward and backward motion fields with respect to the reference frame. However, coding motion vectors in both directions induces complexity and coding overhead, for only a slight quality improvement.

In our scheme, only the backward motion fields are coded and used to infer the forward ones, using an a priori selection strategy. A given block $\mathcal{B}_m$ in frame f_i, is mapped to the best matching block ${\mathcal{B}^\star}_m$ in frame f_{i+1} using only the backward motion vectors. The two following criteria allow to select the best trajectory, (i) the minimum distance to the center of the block, and (ii) the scanning order. In the case where more than one motion vector from the frame $i+1$, point to a block that overlaps with block $\mathcal{B}_m$ in frame f_i, the selection is based on the nearest neighbor criteria. In the low probability case where this criteria is not sufficient to choose the best candidate vector, the scanning order is determinant. The selection of the motion trajectories is represented by the generic motion mapping operator $\mathcal{W}$ as,

$$\mathcal{W}_{i\rightarrow i+1}(I(x,y,i)) \approx I(x,y,i+1) \qquad (1)$$

where $I(x,y,i)$ denotes the samples of frame i in the video sequence I. Figure 2 illustrates the steps involved during the trajectory prediction. The dotted lines correspond to possible paths which are discarded during trajectory prediction.

Now, the 3-D atom is built by replicating its spatial component along the motion trajectory passing by its center in the reference frame, which is chosen dynamically as the frame with the largest energy in the GOP.

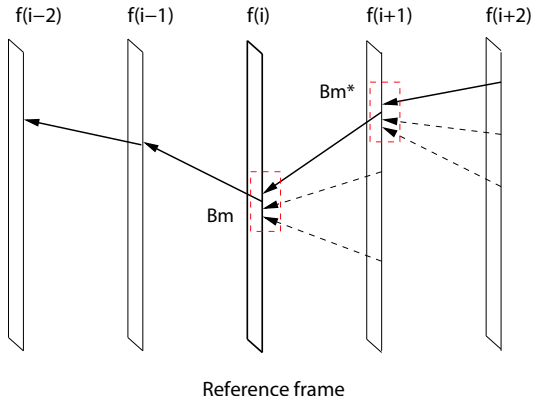

Fig. 2. Example of determining a trajectory for a block in frame i.

C. The Spatio-Temporal Dictionary

The spatial component of the dictionary [8] is generated by applying affine transformations on two mother atoms, which are a 2-D Gaussian $g_1(x,y) = \frac{1}{\sqrt{\pi}} e^{-(x^2+y^2)}$, and its 2^{nd} partial derivative (a ridge-like function) $g_2(x,y) = \frac{2}{\sqrt{3\pi}}(4x^2 - 2)e^{-(x^2+y^2)}$. The 2-D Gaussian is used in order to extract the low frequency components. Its 2^{nd} partial derivative is used to capture image singularities like edges and contours. The affine operator is a composition of translation, scaling and rotation of the mother atoms, as follows:

$$\mathcal{U}_{(x_0,y_0,a_1,a_2,\theta)} g = \frac{1}{\sqrt{a_1 a_2}} g(r_{-\theta}(\frac{x-x_0}{a_1}, \frac{y-y_0}{a_2}))\,,$$

where $r_{-\theta}$ is a rotation matrix of angle θ. The temporal component of the redundant dictionary is spanned by translating and scaling a β-spline $\beta^3(t)$ [7].

D. The MP Decomposition

The 3-D transform consists in computing a decomposition of the GOP in a finite number of spatio-temporal atoms

along motion trajectories, by applying the Matching Pursuit algorithm. A full 3D search is however computationally too complex, as it requires to evaluate $N \times D$ scalar products, where D is the cardinality of the dictionary and N the number of iterations. A heuristic search algorithm is used to reduce the complexity, and it is described by Algorithm 1. The iterative search algorithm first selects the frame with the highest energy, in the GOP $I^n(x, y, i)$, where $I^0(x, y, i)$ represents the original pictures. It then performs an exhaustive spatial search in the selected frame, in order to find the M best candidates among the spatial atoms in the dictionary using an FFT algorithm. Each one of the M candidates[1] is then used to build spatio-temporal atoms, aligned on the motion trajectories according to $\mathcal{W}$, and with the spline temporal functions defined in [5]. The atom which minimizes the energy of the residual signal $I^{n+1}(x, y, i)$ is then selected. The process is then repeated until the signal expansion is long enough, or until a residual error energy threshold has been reached.

Algorithm 1 The Heuristic Search Algorithm.

1: Let $I(x, y, i), i = 1..N_{GOP}$ be a block of frames
2: Select a reference frame r with the largest energy
3: Use the 2D FFT-based exhaustive search algorithm to find the best M uncorrelated candidates among spatial atoms
4: Search for the best 3-D atoms starting from the M candidates, mapped on motion trajectory
5: Update the residual $I^n(x, y, i)$ accordingly and iteratively get back to step 2.

This algorithm has a complexity of order $O(N \cdot (D_s \cdot n \log n + M \cdot n \cdot N_{GOP}))$, where D_s is the cardinality of the spatial dictionary, n is the size of the image, M is the number of candidates, and N is the number of selected atoms.

E. Progressive Coding

The coefficients and atom indexes, where (p_x, p_y, p_t), (a_x, a_y, a_t) and θ, respectively represent the position, the scale and the spatial rotation of the spatial mother functions, have to be scalably encoded in order to provide a bit-rate and resolution-adaptive video signal representation. This operation is fundamental to fully benefit from the intrinsic scalability properties of Matching Pursuit expansions over geometrical dictionaries.

The embedded coding in the MP3D is achieved through the sub-sets partitioning approach [7]. The series of atoms is partitioned into S disjoint sub-sets s_i, where each subset contains l_i elements. These subsets can be seen as energy sub-bands. Their number is dictated by scalability requirements (i.e., the number of target decoding rates), and represents a trade-off between stream flexibility, and coding efficiency, that respectively increases and decreases with S. In each subset, atoms are sorted according to their spatial positions, that are further run-length encoded. The remaining index parameters and quantized coefficients are encoded with the adaptive arithmetic encoder [9]. The resulting bit-stream is piecewise progressive, and optimal truncation points are defined at sub-set limits.

[1] M is chosen to be proportional to the number of blocks in a picture, in this scheme.

III. Experimental Results

In this section, we evaluate the rate-distortion performances of our codec by comparing it with two reference schemes, MPEG4 [10] and H.264 [11]. The standard Foreman and Football sequences in CIF format at 30 fps were used to generate the results. In all experiments, we used a GOP size of 16 (IPPP...for MPEG4 and H.264). It can be seen on Fig. 3 and 4 that the PSNR of MP3D is higher than that of MPEG-4 by about 1-1.5 dB for both sequences and over a wide range of bit-rates. Meanwhile, it is only slightly inferior to the performance of H.264, staying within a 1 dB gap.

Our scheme performs better on the Football sequence for example, where it stays close to H.264 over the whole range of bit rates under consideration. However, we noted that the results for the Foreman sequence always penalize our scheme at high rate. The MP3D does not perform very well for texture, mainly due to the construction of the dictionary (see Fig. 5); once most geometrical information has been encoded, and the PSNR tends to saturate at high rate. Finally, it is noteworthy that both H.264 and MPEG-4 are non-scalable video coding schemes, optimized for compression performance, contrarily to MP3D.

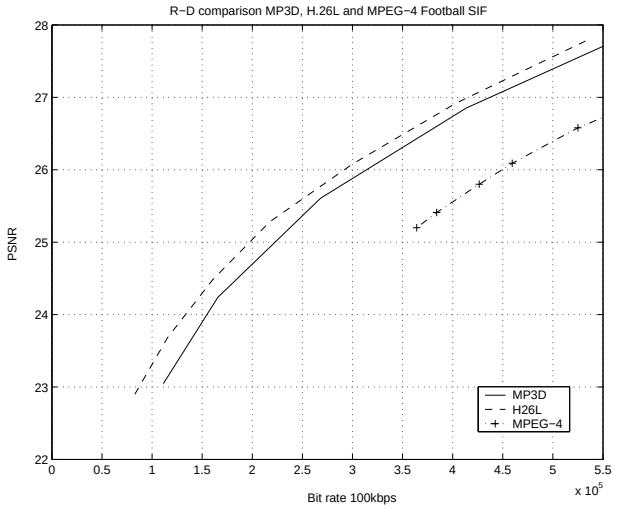

Fig. 3. R-D Comparison of MP3D against h.264 and MPEG-4 for Football

Fig. 5 shows visual comparisons of the first frame from the Football sequence decoded at 550 kbps, using the schemes mentioned before. One can see that H.264 produces more uniform regions. The regions in MP3D are also very smooth, but most prominent edges are well captured due to the nature of the dictionary we used. On the other hand, MP3D lost most of the textures. Overall, MPEG-4 produces a slightly inferior visual quality. Of course these tests are not conclusive, but they allow to emphasize the behavior of MP3D in capturing first the geometrical features in image sequences.

(a) Original

(b) MP3D

(c) H.264

(d) MPEG-4

Fig. 5. Visual Comparison for Frame 1 of Football decoded at 550 kbps

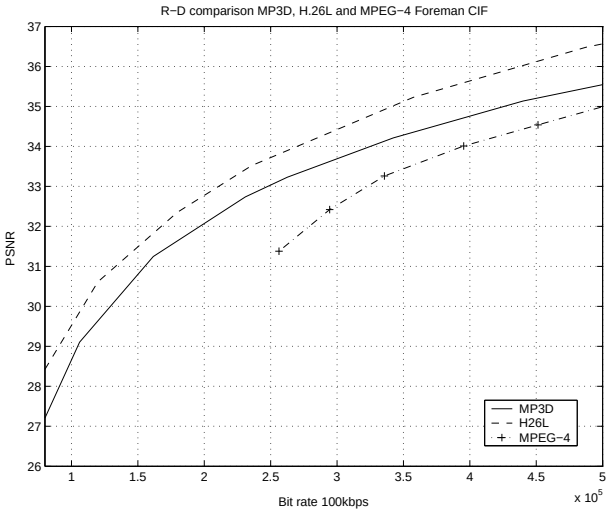

Fig. 4. R-D Comparison of MP3D against H.264 and MPEG-4 for Foreman.

IV. Scalability Properties

The multiresolution structure of the dictionary, the nature of MP, and the embedded coding make the bitstream highly scalable, offering 3-D geometric (i.e. spatio-temporal) and SNR scalability. The geometric properties of the dictionary ensure very easy sequence adaptation prior to decoding. As

Fig. 6. Frame 1 of Foreman decoded in QCIF from the CIF bitstream.

a result, a single bitstream can be decoded at any spatial resolution (as long as the re-scaling is isotropic) and at various frame rates, without resorting to costly re-encoding or post-processing operations. For example, a coded video signal I of spatial size $W \times H$ with a frame rate F can be spatially decoded into a video signal $\tilde{I}$ of spatial resolution $\alpha W \times \alpha H$ at the same frame rate as follows. First the full atom trajectory is reconstructed at the initial size using the motion field operator $\mathcal{W}$. Then each individual atom is analytically re-scaled by simply transcoding its index values (scales and positions) as described in [5]. The new signal becomes :

$$\tilde{I} = \sum_{i=0}^{N-1} \alpha \; c_i \widetilde{\mathcal{W}(g_{\gamma_i})}, \tag{2}$$

where c_i are the atom coefficients and $\widetilde{\mathcal{W}(g_{\gamma_i})}$ corresponds to the motion-mapped atom $\mathcal{W}(g_{\gamma_i})$ after transcoding. We noted that, when transcoding by $\alpha < 1$, thus to a lower resolution, possible aliasing from very small atoms saturate quickly PSNR quality as rate increases. The smallest atoms are thus simply discarded. Figure 6 shows frame 1 of the Foreman sequence decoded in QCIF format from the bitstream corresponding to CIF format. Clearly, the spatial resolution adaptation preserves image features after transcoding. These structures are indeed well captured by our dictionary and the corresponding atoms are simply re-scaled, when decoded at a different resolution. This clearly brings a great advantage in visual quality.

Besides geometric scalability, MP3D provides natural SNR scalability because of the exponential decay of MP coefficients and the embedded quantization. Fig. 7 shows frame 1 of the Foreman sequence decoded at 320 kbps from a bitsream, that was encoded using the multiple rate constraints $\{75, 135, 245, 360, 500\}$ kbps, with an average PSNR of 33.8 db.

Fig. 7. Frame 1 of sequence Foreman decoded at 320 kbps, from the 500kpbs bitstream

V. Conclusions

In this paper, a video coding scheme based on motion-adaptive signal decompositions over a redundant dictionary of waveforms is presented. The over-complete dictionary is designed to model image primitives, mostly edges, that are likely to display coherent trajectories over time. The Matching Pursuit algorithm is used to obtain a compact signal representation. The motion trajectories of prominent image primitives are determined form the motion fields, which are estimated using block matching techniques. A redundant temporal dictionary is also used for temporal decomposition. A progressive bit-stream is generated from the selected atoms using the subsets approach. The compressed video sequence can further be decoded at any resolution due to the parametric structure of the redundant libraries used to represent the information. These geometric stream manipulations are lightweight and can be performed at the decoder or by some simple network intelligence. Comparisons with state-of-the-art codecs illustrate the good performance of the proposed scheme at low bit-rates and motivate its possible use as a base layer in a more general scalable coding framework.

References

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

[2] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on image processing*, vol. 3, pp. 559–571, Sept 1994.

[3] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE transactions on image processing*, vol. 12, no. 12, december 2003.

[4] E. J. Candes and D. L. Donoho, "Curvelets- a surprisingly effective nonadaptive representation for objects with edges," in *Curve and surface fitting*, A. C. C. Rabut and L. L. Schumaker, Eds. Saint-Malo: Vanderbilt University Press, 1999.

[5] A. Rahmoune, P. Vandergheynst, and P. Frossard, "MP3D: A highly scalable video coding scheme based on matching pursuit," in *Proceedings IEEE ICASSP*, vol. 3, Montreal, May 2004, pp. 133–136.

[6] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[7] A. Rahmoune, P. Vandergheynst, and P. Frossard, "Flexible motion-adaptive video coding with redundant expansions," *IEEE Transactions on Circuit and Systems for Video Technology*, 2004, submitted.

[8] P. Frossard, P. Vandergheynst, and R. F. i Ventura, "High flexibility scalable image coding," in *Proc. SPIE VCIP*, Lugano (Switzerland), 2003.

[9] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, vol. 30, pp. 520–540, June 1987.

[10] "MPEG-4 reference software, http://megaera.ee.nctu.edu.tw/mpeg/."

[11] "H.264/AVC reference software, http://bs.hhi.de/ŝuehring/tml/."