FULLY SCALABLE OBJECT BASED VIDEO CODER BASED ON ANALYSIS-SYNTHESIS SCHEME

Chaumont Marc, Cammas Nathalie[1] *and Pateux Stéphane*
IRISA, Campus de Beaulieu, 35042 Rennes, France
Email: Marc.Chaumont@irisa.fr
[1] France Telecom RD, 4 rue du Clos Courtel, 35510 Cesson-Sévigné, France

ABSTRACT

In this paper we present a novel object based video coder. This coder is based on an analysis-synthesis approach which allows for decoupling shape, motion and texture informations. These informations are then coded using wavelets decomposition and progressive coding allowing to have full scalability (object, SNR, temporal, and bitstream scalabilities). Experimental results show the benefits of proposed scheme providing performances close to state of the art video coders while providing scalability.

1. INTRODUCTION

In image or video coding, region-based [1], object-based [2] or model-based [3] coding techniques have been often proposed as ways to improve coding schemes. The main interest of object-based video coding often advanced is content manipulation and object scalability. Manipulation of video content is an important feature of a large amount of multimedias applications. Object scalability allows to allocate more or less bits to different objects of a scene: for example foreground and background in a visio-conference context...
In such schemes, codecs let appear the notion of three different fields of information when coding objects: shape, motion and texture. On the extreme case of model based coding, these notions may be treated separately. Scalable coding schemes may then be obtained thanks to level of details (LOD) such as proposed in VRML.

However, in usual video object-based coder such as MPEG4, these informations are not completely independent. Texture coding relies on shape coding and motion information. Shape coding relies also on motion information. These dependencies then limit scalability features. Moreover in video, coding schemes generally rely on predictive coding techniques. Loss of efficiency are then observed when looking for scalable schemes (e.g. enhancement layers approach of H263, or progressive coding schemes such as MPEG4-FGS).

On the other hand, in the field of image coding, wavelets have been shown to be efficient tools for providing scalable coding schemes (e.g. EZW, SPIHT, JPEG2000, ...). Several works have then proposed to use wavelets for video coding [4], [5]. However these schemes suffer from the motion present in video. If they do not exploit motion, they suffer from poor decorrelation properties. When trying to exploit motion, they suffer from non orthogonality decomposition and strong dependence on motion.

In this paper we then present a novel object-based video coder with full scalable features. This coder relies on a scheme working independently on shape, motion and texture information thanks to an analysis-synthesis approach and wavelet coding techniques. In section 2, we will first present principle of our analysis-synthesis approach. Section 3 and 4 will then present techniques used for coding motion, texture and shape information based on wavelets. Experimental results will be provided in section 5 with comparison to existing coding schemes. Finally section 6 concludes our work.

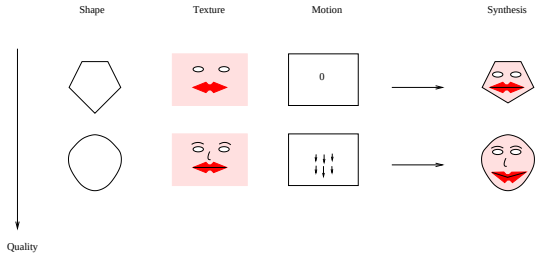


Fig. 1. Scalable video object coding based on progressive shape, texture and motion informations.

2. ANALYSIS-SYNTHESIS APPROACH FOR VIDEO CODING

Most of classical video coding schemes are pixel-based. They perform on a frame per frame basis and use blocks entities to code texture informations and may be limited since they are too focused on the pixel structure. On the opposite model based coding schemes allows for very important compression performance when considering models rather than pixels. However model-based coder suffer from restricted class of application (typically head and shoulders sequences).

A good tradeoff is then to consider object-based video coding coupled with analysis-synthesis approach. After an analysis step shape, motion and texture informations are extracted and defined separately. Shape information can be extracted thanks to segmentation, while motion tracking, thanks to active meshes such as presented in [6], allows to separate motion from texture (see fig. 3 for an example of mesh tracking). Motion, texture and shape informations of each objects may then be coded independently in a scalable way. Moreover mesh tracking benefits from object segmentation which limits mesh degeneracy on occlusion boundaries. Thanks to this new video representation final bit-stream is fully scalable. It can be decoded at different bit-rates and at different qualities for each kind of information: motion, texture or shape and for each object. Figure 1 shows an example of scalable decoding of various information for the rendering of an object.

3. MOTION AND TEXTURE CODING

3.1. Motion estimation and coding

The analysis stage works on group of N frames (GOP) and performs motion estimation between frames using active meshes. Active meshes are good tools for motion compensation, they provide long term continuous tracking of the texture which justify the use of wavelet transform along motion trajectories performed later. Motion estimation is performed as in [6] (see figure 3 for an example of motion tracking between distant frames).

Considering analysis-synthesis scheme with separation between motion and texture, motion can also be lossy coded without significant perceptual distortion (see fig. 2). Hierarchical motion representation is then exploited to this extent [7]. Enhancement informations are lossy coded with progressive bit-planes encoding. An arithmetic coder is used to ensure good efficiency. Compression gain in motion coding can then later be repercuted to texture coding.

(a) original frame

(b) reconstructed frame

(c) DFD

Fig. 2. Example of lossy motion coding for sequence Rue. While visual quality is good for reconstructed frame, DFD is large due to lossy coding of motion information. PSNR of reconstructed frame is 20dB.

3.2. Texture coding

Thanks to motion estimation via mesh tracking, frames are mapped on reference grids, like in [8] or [9] but this time with perfect motion compensation [10]. This step allows to separate motion and texture informations.
Texture frames are then coded using 3D wavelet transform. Temporal transform is then naturally performed along motion trajectories thus best exploiting temporal redundancy. The use of reference grids to represent texture permits to use textures independently from motion during the temporal transform. Motion compensation is needed only if transform is performed between frames that have different reference grids. Thus lossy coding of motion

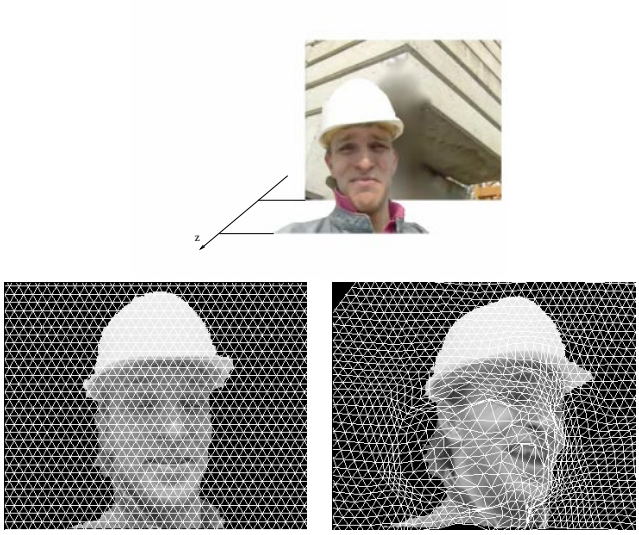


Fig. 3. Motion tracking using active meshes on objects.

refinement can be performed. This separation allows to use perfect motion compensation at the coder while using coarse motion compensation at the decoder in order to ensure low bit-rates. The temporal transform uses the 5/3 lifting filter.
Temporal subbands are further transformed by 2D spatial wavelet transform. Spatio-temporal texture subbands are finally coded using a scalable coder, typically EBCOT.

4. LOSSY CONTOURS CODING

The shape coding method is the one proposed in [11]. Firstly, the object contours are extracted from the segmentation map. Secondly, the successive contours are mapped, aligned and padded. Thirdly, the resulting 1D+t signal is encoded thanks to an IPB scheme and a wavelet decomposition.

4.1. Contours' extraction

A mask and the local object z-order, allow the extraction of a valid contour for an object. A valid contour is the external envelope without the parts due to occultation. Without, loss of generality, we will restrict ourself to the outer contour. Thus, a shape description is a list of positions extracted from the real contour object. An example of such a shape is represented on Fig.4. Breaks are intentionly introduced in our scheme in order to have a more realistic shape definition for objects. They will be further padded in order to estimate complete shape of the object.

Once a position list is obtained for each frame, the lists are motion projected on a reference frame. This treatment let us benefit of temporal consistency. Indeed, the motion compensation helps the mapping process. Further working on projected reference frames rather than separately on each frame allow to effectively separate motion information from shape information.

4.2. Mapping of the contours and contours' alignment

Since shapes' contours will be coded using temporal decorrelation techniques, it is necessary to have a mapping between every

(a) Initial segmentation map

(b) Partial outer contour

Fig. 4. Extraction of the apparent contour of an object

contours along time. If we consider two contours C_t and C_{t+1}, defined at times t and $t+1$, the mapping process tries to map points of C_t and C_{t+1}. Points mapped together then give partial or whole trajectories (depending on occlusion phenomena and local expansions). The partial trajectories will have to be completed. To this purpose, virtual points are added on each contours allowing to obtained a bijective mapping between all consecutive contours.

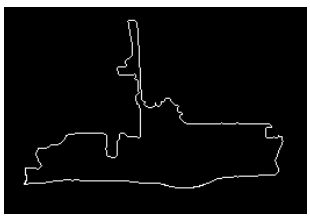
To solve the problem of points insertion the notion of universal abscissa is introduced. Each contour is mapped on an universal abscissa ; this means that each point of each contour own a corresponding value named "the universal abscissa". Once each contours own a map to the universal abscissa, virtual points are then added everywhere a universal abscissa value is missing.(see [11] for details)

4.3. Contours' padding

The spatio-temporal padding is used to closed each contours in the case of "breaks". Thus, the padding objective is to extend continuously each contour.

In a first step, we will add "virtual" points to merge broken contours. In a second step, we will fill "virtual" points by giving them a position obtained by the computation of contours padding (see Fig.5).

(a) Before padding

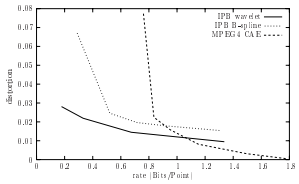


(b) After padding

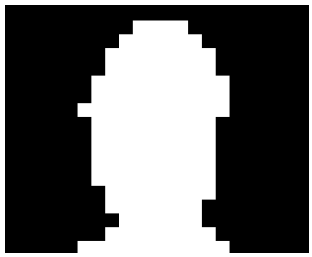
Fig. 5. Spatio-temporal padding illustration

4.4. IBP coding scheme of object shape

Thanks to the re-parameterization of contours on a universal abscissa, contours can then be considered as a kind of 1D+t signal. Decorrelation is then performed using IBP scheme in temporal dimension and wavelet decomposition along abscissa dimension. The first contour of a GOP will be coded intra (I) and others will be coded using a simple prediction (P) or a bidirectional prediction (B). We have considered only one B frame between two successive I or P frames in our experiments but higher number of frame could also be considered.



Performance comparison



CAE R=0.76 Bits/Point

B-spline R=0.72 Bits/Point

Wavelet R=0.67 Bits/Point

Fig. 6. Comparison of contour coding techniques. Average percentage of erroneous pixels with respect to average bit-rate per contour element. Sequence Foreman.

In order to have a hierarchical representation of the contours and to provide enhanced scalability, dyadic wavelet decomposition is performed along universal abscissa. Before performing the wavelet transformation we re-sample the contours in order to get a length equals to a multiple of a power of two. This enables successive circular wavelet decompositions as long as there is just one coefficient left. For this decomposition we use 7/9 Antonini's wavelet filters [12].

Coefficients obtained after this spatio-temporal decomposition are then coded with a bit-plane arithmetic coder. Lossy coding is obtained by choosing the encoded number of bit-planes. Figure 6 presents results obtained with our proposed schemes and compare it to existing schemes such as B-spline coding technique and CAE used in MPEG4. Our proposed technique outperforms other techniques while providing good scalability features.

5. RESULTS

We have tested our proposed coding scheme on several video sequences. Results are presented here for sequences Foreman CIF 15Hz and Erik CIF 15Hz and for very low bit-rates (i.e. below 100 Kbit/s). When considering MPEG4, such very low bit-rates can't be achieved. Effectively, considering foreground object in Foreman sequence, minimal attainable bit-rate is 96 Kbit/s using coarsest quantization parameters (172 Kbit/s in full frame coding mode).

On figure 7 are reported decoded frames for our scheme and H26L. Visual quality is higher for our proposed scheme with more detailed texture and no blocking artifacts nor blur (which is not the case for H26L due to its intensive de-blocking filter). Bit-rate

repartition are provided in tables 1 and 2 as well as PSNR for foreground objects. Further, compared to H26L, our scheme provide scalable features.

6. CONCLUSION

We have presented a novel full scalable video coding scheme. This scheme relies on an analysis-synthesis scheme which allows to decouple shape, motion and texture information. These informations are then later coded using wavelet and efficient progressive coding tools (e.g. bit-plane coding and EBCOT). First experiments show results close to state of the art video coder while providing scalability.

(a) MPEG4

(b) Proposed scheme (c) H26L VM 8

Fig. 7. Comparison between proposed coding scheme and state of the art H26L coder VM 8.4 (2 B frames, CABAC, 5 reference frames, full RD optimization). Foreman sequence, CIF - 15Hz. Erik sequence, CIF - 15Hz.

	Proposed scheme	H26L
Motion (kbs) Texture (kbs) Shape (kbs)	17.56 69.12 6	
Background (kbs)	16	
Total bitrates (kbs)	108	98
PSNR(dB)	32.06	32.59

Table 1. Bitrates repartition for sequence Foreman: object 1: foreground (motion, texure and shape). object 2: background. PSNR are calculated on foreground reconstructed pixels for the proposed scheme and for H26L

	Proposed scheme			H26L	
Motion (kbs) Texture (kbs) Shape (kbs)	11.9 28.6 3	12.23 37.8 3	12.25 47.4 3		
Background (kbs)	11.5	11.5	11.5		
Total bitrates (kbs)	55	64	74	52	70
PSNR(dB)	27.9	28.4	29	26.9	28.2

Table 2. Bitrates repartition for sequence Erik: object 1: foreground (motion, texure and shape). object 2: background. PSNR are calculated on foreground reconstructed pixels for the proposed scheme and for H26L

7. REFERENCES

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–574, Apr. 1985.

[2] ISO. Draft MPEG-4, "Video verification model version 8.0.," *ISO/IEC JTC1/SC29/WG11*, 1997.

[3] D. E. Pearson, "Developments in model-based video coding," *Proc. IEEE*, vol. 83, pp. 892–906, June 1995.

[4] J. Vieron, C. Guillemot, and S. Pateux, "Motion compensated 2d+t wavelet analysis for low rate FGS vido compression," *International Thyrrhenian workshop on digital communications 2002 (invited paper)*, Sept. 2002.

[5] 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, february 1999.

[6] S. Pateux, G. Marquant, and D. Chavira-Martinez, "Object mosaicking via meshes and crack-lines technique. application to low bit-rate video coding," *Picture Coding Symposium, PCS'2001*, Apr. 2001.

[7] G. Marquant, S. Pateux, and C. Labit, "Mesh and "crack lines": Application to object-based motion estimation and higher scalability," *IEEE International Conference on Image Processing ICIP 2000*, vol. 2, pp. 554–557, Sept. 2000.

[8] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, Sept. 1994.

[9] Heiko Schwarz and Erika Müller, "Object-based 3-d wavelet coding using layered object representation," *IEEE International Conference on Image Processing, ICIP'2000*, vol. 1150, Sept. 2000.

[10] Natalie Cammas and Stéphane Pateux, "Fine grain scalable video coding using 3d wavelets and active meshes," *IS&T/SPIE's 15th Electronic Imaging Science and Technology - SPIE'2003*, vol. 5022, no. 44, Jan. 2003, France Telecom and IRISA.

[11] Marc Chaumont, Stéphane Pateux, and Henri Nicolas, "Efficient lossy contour coding using spatio-temporal consistency," *Picture Coding Symposium, PCS'2003*, pp. 289–294, Apr. 2003.

[12] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205–220, Feb. 1992.