OBJECT-BASED VIDEO CODING USING A DYNAMIC CODING APPROACH

Marc CHAUMONT, Stéphane PATEUX and Henri NICOLAS
IRISA, Campus de Beaulieu, 35042 Rennes, France
Email: Marc.Chaumont@irisa.fr

ABSTRACT

In this paper we propose a video-object based coding scheme using dynamic coding. The principle of dynamic coding is to set on competition different coders on each video object. Thus, we are proposing a video-object based dynamic coding scheme using four completely different coders. The novelty of our work firstly comprise a global rate-distortion optimization enabling an optimal selection of a coder and its parameters for each object, and secondly the definition of a distortion metric. Our work is thus confirms that dynamic coding is efficient. It shows that a video object based coding approach is competitive. It improves object based video coders such as MPEG4 and it gives interesting comparison results between different state-of-the-art coders.

1. INTRODUCTION

Recently, H264/AVC [1] has shown an impressive improvement in video coding efficiency. As a consequence, many alternative approaches are now outdated because they are no more enough efficient (for example, region-based approaches are no more competitive). What about object-based video coding? It is clear that object-based video coding may have an interest for functionalities such as bit-rate repartition, video editing, and potentially for better motion estimation, but is it as efficient as the H264/AVC coder in terms of compression performances?

This paper shows that object video coding is still competitive if it is used in the context of a dynamic coding approach. It also allows improvement of video object coding approaches such as MPEG4 [2]. One starts with this observation: the best video object coder is not always the same on each object and the coding performances depend on video objects properties (motion, textures, statistics etc). During the MPEG4 normalization process in 1995, dynamic coding was proposed. It is based on the selection of the best coding technique among a set of candidates [3], [4].

Our goal is then to propose a dynamic coding scheme for an object based video coding approach. To reach this objective, we put in competition state of the art coders which have different ways of representing data and whose performances depend on the objects' characteristics. Section 2 presents these coders.

We should notice that some questions occur when different coders are used together. Which common distortion metric is used? How are bit-rates and coders attributed to objects given a global constraint (quality constraint or bit-rate constraint)? Section 3 considers those rate-distortion aspects.

Section 4 gives some results, and a conclusion is provided in Section 5.

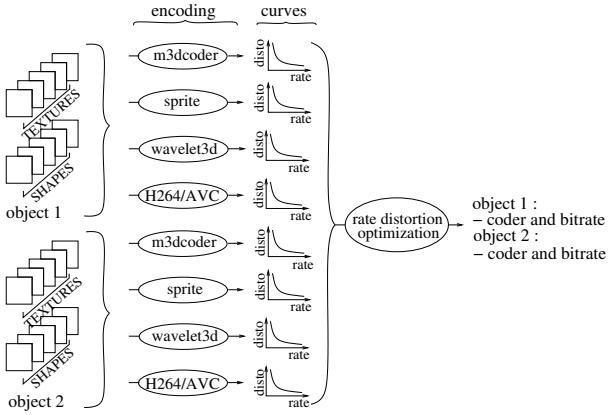


Fig. 1. Video-object based dynamic coding scheme (2 objects).

2. CODING SCHEME

Our object-based dynamic coding scheme assumes that object's masks are known for a given sequence.

The general scheme of the proposed method is illustrated in Figure 1. Texture and motion information of each object is encoded at different bit-rates with different coders. Then, thanks to the rate-distortion response curves, a rate-distortion optimization (see Section 3) assigns a bit-rate and a coder for each object given a global constraint (in terms of quality or bit-rate). Note that shape information is coded separately through a lossy shape coding method [5].

The next sub-sections present the different coders used in our video-object based dynamic scheme.

2.1. 3D model-based coder

The 3D model-based coder (*m3dcoder*) [6] is based on an analysis-synthesis scheme. During the analysis step, a rigid 3D model and camera positions are computed on each GOP[1]. The synthesis step uses the 3D model, the camera positions and a unique image (called the reference frame) to rebuild a GOP.

In the coding part, *m3dcoder* codes for each GOP one 3D model, camera positions and one reference frame. For that purpose, 3D models are represented as a uniform mesh whose vertices are quantized and then encoded with JPEG2000 [7]. Camera positions are differentially coded. The reference frames are coded using JPEG2000 (Intra coding for the first GOP and Inter coding

[1]GOP: Group Of Pictures.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

for the other GOP). The coding process is achieved with a rate constraint that imposes a bit-rate constraint per GOP.

2.2. Sprite coder

The sprite coding approach is a very common approach already proposed in MPEG4 [2], which has shown its relevance in a recent Object-Based Analysis-Synthesis Coder (OBASC) [8]. The objective is to build a unique picture summarizing textures appearance (typically for background objects). For that purpose, a parametric motion should be computed.

We propose to use the motion estimator based on mesh estimation and to build a sprite similarly to [9]. Once the sprite is built, a texture padding is applied and this picture is encoded using JPEG2000 [7]. Motion is simplified into an affine motion model thanks to robust parameter estimation. The six parameters are then differentially encoded from frame to frame.

2.3. 2D+t wavelet coder

The 2D+t wavelet coder (*wavelet3d*) is used in its object mode as explained in [10]. It is a scalable coder and its particularity is to perform scalability independently on each of the three informations : texture, motion and shape, thanks to an analysis-synthesis approach.

The principle of this coder is to project a group of frames onto one or several reference frames, in order to de-correlate textures and motion, and then to perform a spatio-temporal wavelet decomposition.

For each GOP, a 5/3 lifting filter is applied along the time axis. Resulting sub-bands are spatially decomposed with a 9/7 Daubechies filter [11] and finally encoded with EBCOT [12].

In a similar way, motion is temporally and spatially transformed. Temporal decorrelation uses a 9/7 Daubechies filter and spatial decorrelation performs a pyramidal decomposition taking into account the mesh structure. Sub-bands are then encoded with a bit-plane arithmetic coder.

2.4. H264/AVC object based coder

The H264/AVC coder [1] is currently the most efficient block based coder. Its technical design was completed in December 2002 inside the ITU and is now incorporated in MPEG4 standard. Many improvements have been done in comparison to oldest standards. For example, authors of [13] give a mean bit-rate earning of 39% on MPEG4 ASP, 49% on H263 HLP and 64% on MPEG2.

To reach better coding efficiency, we have chosen H264/AVC to encode objects instead of MPEG4. This however requires a modification of the H264/AVC coder to adapt it to video object based coding. To that extent, we only encode the useful macroblocks covering the video object. Notice that texture padding is processed on each VOP[2] before macroblock coding. With that modification the decoding step is then shape independent. Indeed, we do not need shape information at the decoding process (the only additional information is the – coded or non-coded – state information for each macroblock).

3. RATE-DISTORTION OPTIMIZATION

Now that the four shape independent coders have been presented, we consider the choice of a common distortion metric and the rate-distortion optimization step used to perform object-based selection of the best coders.

3.1. Distortion metric

This sub-section deals with the definition of a distortion metric. This is an important issue when dynamic coding scheme is involved. Indeed we are putting into competition coders that are producing very different kinds of artifacts or distortion. Then, it is necessary to specify a fair metric before making any comparison.

The definition of distortion metrics is something that has been largely studied. For example the VQEG [3] group tries to define novel quality metrics by emulating the Human Visual System. At present, no novel metrics really show significant performances. In consequence, the PSNR [4] metric (deduced from MSE[5] metrics) is often retained. Moreover it is a well known metric used in the standard coder evaluation.

The main drawback of the PSNR metric is that it does not take into account geometrical distortions. As a consequence the metric that we have chosen is what we are calling the *PSNR in the texture domain* and noted as: PSNR_{text}. This metric is the classical PSNR when there is no projection and decorrelation between textures and motion. Thus for the case of H264/AVC object coder we will keep the traditional PSNR metric. For *wavelet3d*, *m3dcoder* and sprite coders, our metric is a PSNR_{text} computed between projected textures and projected coded-decoded textures. This metric is thus invariant with respect to motion errors. Equivalently to PSNR_{text}, we define *MSE in the texture domain* which is noted as: MSE_{text}. By using PSNR_{text} (or equivalently MSE_{text} metric), we believe that the comparison between each coder will be fairly and visually more realistic.

3.2. Rate-distortion optimization

Rate-distortion optimization takes place after the generation of rate-distortion curves for each object and each coder (note that the distortion metric used is the MSE_{text}). This implies that each object has been coded-decoded at different bit-rates for each coder.

The rate-distortion optimization objective is to distribute bit-rates among the different objects under a global constraint and to choose the best coder for each object.

Rate-distortion optimization can be written as the minimization of the distortion D under the constraint that rate R is below the global constraint R^* (see equation 1). Its objective is to find the best set of points $\{p^*_{o,c,i}\}$ belonging to the rate-distortion curves $\mathcal{C}$. A point $p_{o,c,i}$ defines the i^{th} rate-distortion point on the rate-distortion curve ($p_{o,c,i} = \begin{pmatrix} R_{o,c,i} \\ D_{o,c,i} \end{pmatrix}$) for the coder c and the object o. The solution is a set $\{p^*_{o,c,i}\}$ with a unique point $p_{o,c,i}$ for each object o such that:

$$\{p^*_{o,c,i}\} = \arg \min_{\{p_{o,c,i}\} \in \mathcal{C}} D : R \leq R^*, \quad (1)$$

$$with: \quad D = \Sigma_o D_o(p_{o,c,i}),$$
$$and: \quad R = \Sigma_o R_o(p_{o,c,i}).$$

[2] VOP: Video Object Plane.

[3] VQEG: Video Quality Experts Group.

[4] $PSNR = 10 \log(\frac{255^2}{MSE})$ for a grey level image coded on 8 bits.

[5] MSE: Mean Square Error.

Equation 1 could be transformed into a non constrained minimization with the use of a Lagrangian. We defines a Lagrangian functional $J_\lambda(\{p_{o,c,i}\})$ which has to be minimized over each curves' points given λ (iterations are performed in order to obtain a λ whose value allows the global constraint R^* to be reached):

$$\begin{aligned} J_\lambda(\{p_{o,c,i}\}) &= D + \lambda R \\ &= \Sigma_o[D_o(p_{o,c,i}) + \lambda R_o(p_{o,c,i})] \end{aligned}$$

4. EXPERIMENTAL RESULTS

Experiments of our video-object based dynamic coding scheme have been done on the following sequences: *Foreman* (CIF 15Hz), and *Thabor Stairs*[6] (CIF 25Hz). The *Foreman* segmentation is generated from a manual labeling of spatial regions. Rate-distortion curves are given in Figures 2 and 3.

When comparing traditional H264/AVC (i.e full frame) and our video-object based dynamic coding scheme for bit-rate around 100Kb/s, results are better for the object approach in both the terms of the PSNR$_{text}$ metric and the visual reconstruction (see Figures 4 and 5).

The bit-rate distribution of Table 1, shows that in the *Foreman* sequence, 81% of the total bit-rate is devoted to the encoding of the foreground object and only 13% to the encoding of the background. The 6% remaining bit-rate is used for shape coding. Thus, dynamic coding allows to give more bit-rate to the objects that are not temporally stable i.e with strong luminosity variation, abrupt or non rigid motion, and auto-occultation.

At very low bit-rates, our video-object based dynamic coding scheme can perform better than H264/AVC. However, at upper bit-rate (more than 250 Kb/s for a CIF 15Hz sequence), gains obtained by using an object approach are not strong enough to compensate object overhead (shape, texture and description overhead); H264/AVC is then better (in terms of PSNR$_{text}$) than video-object based dynamic coding. As an example, Table 2 gives a rate-distortion comparison where our dynamic scheme is not as good as H264/AVC. Nevertheless, results are visually similar.

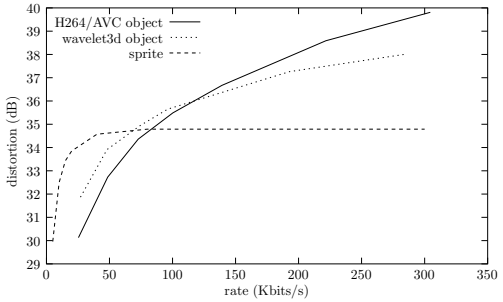
5. CONCLUSION

In this paper we proposed a video-object based dynamic coding scheme which puts into competition four state-of-the-art coders: a model based coder (*m3dcoder*), a sprite coder, a 2D+t wavelet coder (*wavelet3d*) and an object-based H264/AVC coder. For that purpose a novel distortion metric has been defined (MSE$_{text}$), which allows to generate rate-distortion curves. After processing rate-distortion optimization, we assign a coder and a bit-rate to each video object.

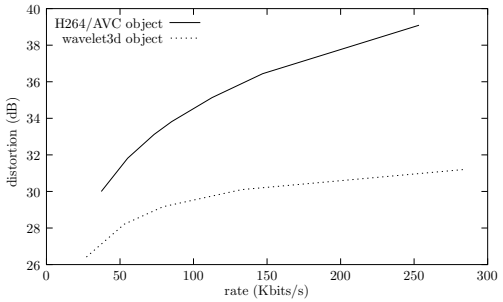
The results obtained with our proposed video object dynamic coding are better than H264/AVC at very low bit-rate (around 100 Kb/s for a CIF 15Hz sequence) and at low bit-rate (around 250 Kb/s for a CIF 15Hz sequence), results are visually similar.

Since dynamic coding is highly CPU consuming, an improvement would be to replace the time computational consuming steps, i.e extraction of the rate-distortion curves and rate-distortion optimization, by a simple prediction step like in [14].

[6]The *Thabor Stairs* sequence is a hand film sequence recorded by a walking person. Let's note that the scene content is rigid.



(a) background RD curves



(b) foreground RD curves

Fig. 2. *Foreman* CIF 15Hz sequence: rate-distortion curves for background (a) and foreground (b) objects on 60 images.

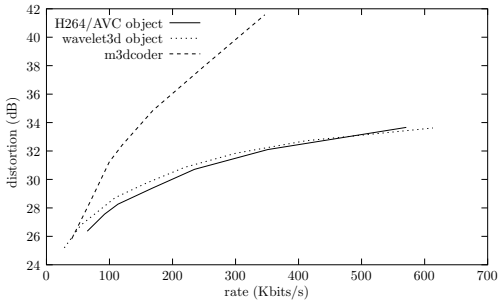
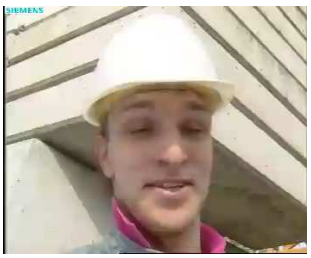


Fig. 3. *Thabor Stairs* CIF 25Hz sequence: rate-distortion curves on 110 images.

(a) Dynamic Coding



(b) H264/AVC non object

Fig. 4. *Foreman* CIF 15Hz sequence: image 1 coded by our video-object based dynamic coding scheme (R=99Kb/s PSNR$_{text}$=33.4) and by H264/AVC non object (R=100Kb/s PSNR=32.9). Bit-rate repartition is given in Table 1. Note that the logo video object is not present but would cost less than 1Kb/s.

(a) m3dcoder (b) H264/AVC non object

Fig. 5. *Thabor Stairs* CIF 25Hz sequence: image 59 coded by our video-object based dynamic coding scheme (R=100Kb/s PSNR$_{text}$=31.3) and by H264/AVC non object (R=113Kb/s P-SNR=28.3).

Table 1. *Foreman* CIF 15Hz sequence at very low bit-rate ($\simeq$100Kb/s): bit-rate repartition for video-object based dynamic coding and comparison with H264/AVC JM5 coding (one B frame, full optimization, Hadamard transformation, CABAC, 5 reference frames).

	Dynamic Coding	H264/AVC
foreground object:	**H264/AVC object**	
rate:	80 Kb/s	
PSNR$_{text}$:	33.54	
shape coding:	**IPB Wavelet**	
rate:	6 Kb/s	
background object:	**sprite**	
rate:	13 Kb/s	
PSNR$_{text}$:	33.22	
rebuilded sequence:		
rate:	99 Kb/s	100 Kb/s
PSNR$_{text}$:	33.4	32.9

Table 2. *Foreman* CIF 15Hz sequence at low bit-rate ($\simeq$260Kb/s): bit-rate repartition for video-object based dynamic coding and comparison with H264/AVC JM5 coding (one B frame, full optimization, Hadamard transformation, CABAC, 5 reference frames).

	Dynamic Coding	H264/AVC
foreground object:	**H264/AVC object**	
rate:	147 Kb/s	
PSNR$_{text}$:	36.4	
shape coding:	**IPB Wavelet**	
rate:	5 Kb/s	
background object:	**WLT 3D**	
rate:	108 Kb/s	
PSNR$_{text}$:	35.9	
rebuilded sequence:		
rate:	262 Kb/s	268 Kb/s
PSNR$_{text}$:	36.2	37.6

6. REFERENCES

[1] T. Wiegand and G. Sullivan, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264—ISO/IEC 14496-10 AVC)," Draft ISO/IEC 14496-10:2003(E) - Draft ITU-T Rec. H.264(2003 E), ISO/IEC MPEG and ITU-T VCEG, Pattaya, Thailand, Mar. 2003.

[2] JTC1/SC29/WG11 Coding Of Moving Pictures & Audio ISO/IEC, "MPEG4 overwiew - (V.21 Jeju version)," Final draft, International Organisation For Standardisation ISO/IEC, Mar. 2002.

[3] T. Ebrahimi and al., "Dynamic coding of visual information," Technical report ISO/IEC JTC1/SC29/WG11/M0320, ISO/IEC, Oct. 1995.

[4] E. Reusens, T. Ebrahimi, C. Le Buhan, R. Castagno, V. Vaerman, L. Piron, C. de Sol Fbregas, S. Bhattacharjee, F. Bossen, and M. Kunt, "Dynamic approach to visual data compression," in *IEEE Transactions on Circuits and Systems for Video Technology*, Feb. 1997, vol. 7, pp. 197–211.

[5] M. Chaumont, S. Pateux, and H. Nicolas, "Efficient lossy contour coding using spatio-temporal consistency," in *Picture Coding Symposium, PCS'2003*, Apr. 2003, pp. 289–294.

[6] R. Balter, L. Morin, and F. Galpin, "Very low bitrate compression of video sequence for virtual navigation," in *Picture Coding Symposium, PCS'2003*, Apr. 2003, pp. 305–308.

[7] ISO/IEC JTC1/SC29 WG1, "Jpeg2000 coding of still pictures," Final Committee Draft 15444-1, ISO/IEC, Apr. 2000.

[8] S. Okada, K. Jinzenji, and N. Kobayashi, "An approach to MPEG-4 multi mode coding using sprite coding," in *International Picture Coding Symposium, PCS'2001*, Apr. 2001, pp. 421–424.

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

[10] M. Chaumont, N. Cammas, and S. Pateux, "Fully scalable object based video coder based on analysis-synthesis scheme," in *International Conference on Image Processing, ICIP'2003*, Sept. 2003.

[11] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," in *IEEE Trans. Image Processing*, Feb. 1992, vol. 1, pp. 205–220.

[12] D. Taubman, "High performance scalable image compression with EBCOT," in *IEEE Transactions On Image Processing*, July 2000, vol. 9, pp. 1158–1170.

[13] H. Schwarz and T. Wiegand, "The emerging JVT/H.26L video coding standard," in *International Broadcasting Convention, IBC'2002*, Amsterdam, NL, Sept. 2002, invited paper.

[14] P. Fleury, *Dynamic Scheme Selection in Image Coding*, Ph.D. thesis, École Polytechnique Fédérale de Lausane, July 1999.