EFFICIENT LOSSY CONTOUR CODING USING SPATIO-TEMPORAL CONSISTENCY

Chaumont Marc, Pateux Stéphane and Nicolas Henri
IRISA, Campus de Beaulieu, 35042 Rennes, France
Email: Marc.Chaumont@irisa.fr

ABSTRACT

This paper deals with the problem of lossy video object shape coding. This problem belongs to the area of video object-based coder. Our approach proposes to exploit spatio temporal correlation. This is done by considering the contour evolution and dealing with occultation phenomenon. The novelty lies in a fine mapping of consecutive contours and in the contour padding in occulted parts. Once a group of contours is processed, wavelet transformations with a predictive IPB (Intra, simple Prediction, Bidirectional prediction) scheme is used to encode it. Experimental results show significant improvement at very low bit rate. Moreover, the bitstream is fully progressive.

1. INTRODUCTION

In object-based video coder such as in MPEG4 [1], shape information has to be coded. Efficient coding techniques have been studied to decrease the bit-rate overhead of shape [2][3]. We generally classify those shape coding techniques in two categories, the bitmap-based coding or the contour-based coding.

The bitmap-based coding approaches (MMR: Modified-Modified Read, CAE: Context-Based Arithmetic Encoding, quad-tree...) rely on a bileveling coding technique. As an example, CAE technique [4], uses an arithmetic technique to code each pixel knowing their surrounding context. Some of these approaches take into account temporal redundancy, for instance MPEG4 has developed a solution based on hierarchical BAB (Binary Alpha Blocks), CAE technique and temporal prediction. MPEG4 CAE is the present reference solution for lossy shape coding. The problem of this solution is bloc artifacts at low bit-rate, which is inherent to the adopted modeling.

On the other side, contour-based coding approaches (Freeman, polygonal, B-spline...) only describe the contour. Freeman technique codes a chain of directional displacements. Polygonal and B-spline approach define contours by means of lines segments and respectively control points.

Yoshida and al. [5] have proposed a lossy contour based solution equaling MPEG4 CAE's performance at low bit rate. They took into account the temporal information by simultaneously coding a group of consecutive contours belonging to the same video object. Our approach is inspired by their solution. The main novelty is that we are improving temporal stability by working only with real shape contour. This is done by taking into account contour's part which really belong to the object (see Fig.1). Further, a better temporal mapping is proposed exploiting motion compensation. Thus, we could deals with non rigid object.

(a) Initial segmentation map (b) Partial outer contour

Fig. 1. Extraction of the apparent contour of an object

2. EXTRACTION AND ALIGNMENT OF A CONTOUR BELONGING TO A GROUP OF FRAME

Our objective is to obtain for all contour's points, the corresponding position all along time in the purpose of extracting the evolution surface of the contour. This surface could be represented by two spatio-temporal planes giving a point position (x, y) knowing indices s on the contour and knowing the frame number t (see Fig. 8).

Thus, a bijective mapping of consecutive contours along time should be found. The major problem is that the contour of a video object do not have a constant length throughout time. This is due to deformations, occultation with other objects or image borders. As consequence the contour points may be mapped to zero, one or several points. To solve this problem, points must be artificially added in such a way that all points can be mapped one to one.

The different steps to reach our objective are first the contour extraction and the definition of a list of positions for each contour belonging to the group of frame, second, an adapted mapping between consecutive contours, third, the contours alignment thanks to the computation of an universal abscissa and finally, a spatio-temporal padding to fill missing positions.

2.1. Contour extraction of a group of frame

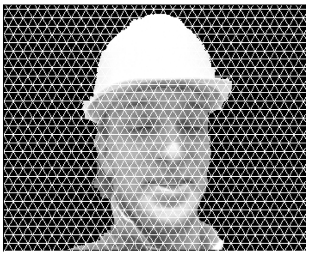
A shape is composed of outer and inner contours. An outer contour is the video object external envelope without the parts due to occultation. Figure 1 shows the outer contour of the little boat of frame 50. Without loss of generality, we will restrict ourself to outer contours.

The knowledge of valid parts (segment which is not due to an occultation) of the external envelope of an object is deduced from the z-order associated to each segment composing the external envelope [6].

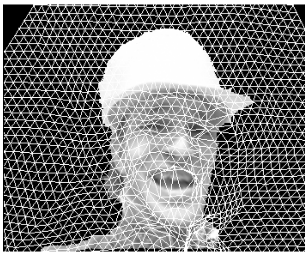
Thus, the shape description is a list of positions extracted from the real contour object (outer contour). This representation possibly shows “break” in the contour e.g. the contour could be open.

Once each positions list is obtained (one list per 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 of sect.2.2. Moreover, in a video object-based coder, it may be interesting to decorrelate motion and contours, and to code each one separately.

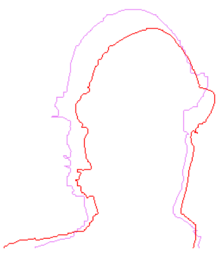
Figure 2(c) shows in red color the contour of frame 8 and in purple color the contour of frame 0. Figure 2(d) shows in red color, the motion compensated contour of frame 8, through the reference frame 0, and in purple color, the contour of frame 0. Motion estimation is obtained thanks to an active mesh [7]. The mesh is initialized on frame 0 (Fig. 2(a)) and is tracked up to frame 8 (Fig. 2(b)).



(a) Mesh on frame 0



(b) Mesh on frame 8



(c) C_8 (red) and C_0 (purple)

(d) $C_8^{displaced}$ and C_0

Fig. 2. Motion compensation of contours

2.2. Adapted mapping of a contour between two frame

If we consider a contour 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}. This is done by looking for couples of points (P_1, P_2) from contour at t and $t+1$ that are at minimal Euclidean distance e.g:

$$P_1 = argmin_{\{P \in C_t\}} dist(P, P_2) \quad (1)$$

By following simultaneously the two contours, a mapping of C_t toward C_{t+1} is found. Similarly, a mapping of C_{t+1} toward C_t is obtained.

An initial bidirectional mapping is deduced and give the couples of points that “links” the contour C_t and C_{t+1}. This bidirectional mapping does not “link” every points. An improvement is then performed by adding few “links” such that between two consecutive “links” there is at most, one contour having non linked points. The resulting mapping is called the adapted mapping. Fig.3 shows two different situations obtained.

The contour at time t and $t+1$ is represented as two oriented graphs. Indices i represent points of contour at time t, and indices j represent points of contour a time $t+1$. The “links” between the two graphs show the mapped point. A specific case when a contour is not closed is also shown.

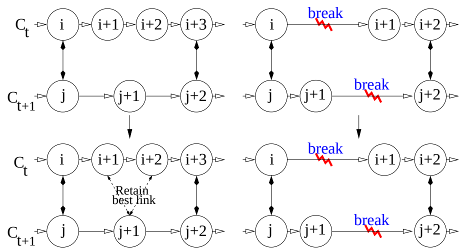


Fig. 3. Adapted mapping (before and after bidirectional mapping improvement)

2.3. Alignement of the contour belonging to a group of frame

The points mapped together give partial or whole trajectories along time. The partial trajectories will have to be completed. To this purpose, virtual points are added on the group of contours to obtain a bijective mapping between all the consecutive contours belonging to the group of frame Fig.4.

To solve the problem of points insertion the notion of universal abscissa is introduced. Each contour is mapped on an universal abscissa. It is an integer number given to each position. This number is the same for points which belong to the same trajectory. It is in increasing order during a contour following. Its definition domain is limited and fully covered. Once each contour owns a map to the universal abscissa, virtual points are added everywhere a universal abscissa value is missing (see Fig.4).

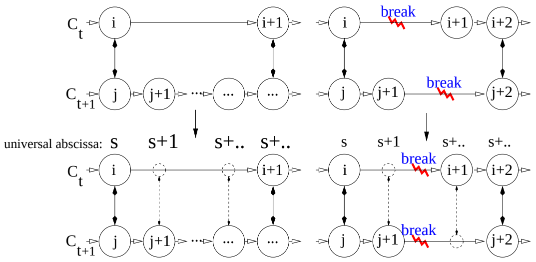


Fig. 4. Contours alignment and virtual points introduction. Virtual points are circles in dot line

2.4. Padding of the contour belonging to a group of frame

The universal abscissa mapping lets appear “breaks” in the trajectories. Those “breaks” have to be filled, and this is done by a spatio-temporal padding. The padding objective is to extend continuously a signal. Here, the signal may be composed of a contour evolving in time.

In a first step, we will add “virtual” points to merge broken contours. Fig.5 shows the addition of points when there is a contour break. Those “virtual” points will be represented by Ω_{Out} set whereas original points will belong to Ω_{In} set.

In a second step, we will fill “virtual” points by giving them a position obtained by the computation of contours padding. Fig. 6 shows the computation evolution of an average signal used to prolong the group of contours.

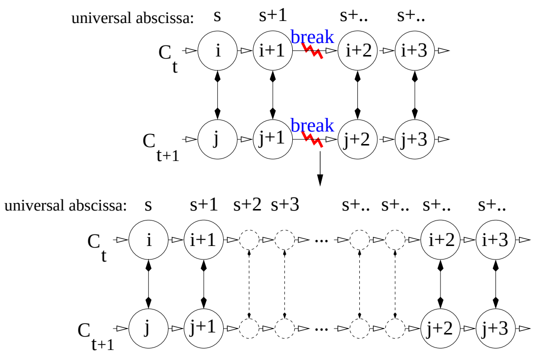
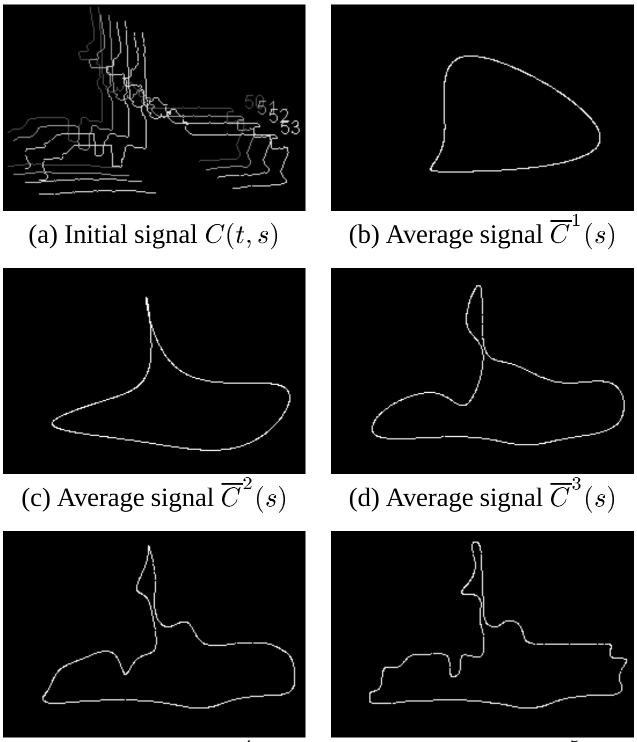


Fig. 5. Virtual points adding to merge broken contours

A contour could be modeled by a parametric representation with a sum of finite function $C(s) = \sum_{k=1}^{k=K} c_k \phi_k(s)$



(a) Initial signal $C(t,s)$ (b) Average signal $\overline{C}^1(s)$

(c) Average signal $\overline{C}^2(s)$ (d) Average signal $\overline{C}^3(s)$

(e) Average signal $\overline{C}^4(s)$ (f) Average signal $\overline{C}^5(s)$

Fig. 6. Average signal $\overline{C}^L(s)$ at different level L

(e.g. B-splines), s being our universal abscissa. This representation, could be generalized to represent a contour evolution with $C(t,s) = \sum_{k=1}^{k=K} c_k \phi_k(t,s)$, t being time.

Thus, to prolong a signal (for us, a group of consecutive contours belonging to a same video object), the idea is to find a parametric signal $\overline{C}^L(s)$ which is identical where the original signal is defined and which is a smooth extension elsewhere. This signal will be used to closed the group of contours representing the object shape evolution (see Fig.7). Remark that $\overline{C}^L(s)$ is not time dependent. This is justified by the contour temporal stability obtained thanks to the motion compensation of contours (see sect.2.1).

We propose for $\overline{C}^L(s)$ a hierarchical representation with:

$$\overline{C}^L(s) = \sum_{l=1}^{L} \Delta C^l(s) \tag{2}$$

and

$$\Delta C^l(s) = \sum_{k=1}^{k=2^{l+1}} \delta c_k^l \phi_k^l(s) \tag{3}$$

ϕ_k^l being multi-scale functions. Coefficients δc_k^l are then estimated using successive analysis and synthesis steps exploiting the hierarchical representation of $\overline{C}^L(s)$. Thus,

$\overline{C}^L(s)$ is iteratively refined going from coarse to fine levels (see Fig. 6).

The synthesis filtering allows to update the average signal $\overline{C}^l(s)$ of level l thanks to the refinement signal $\Delta C^{l-1}(s)$. The refinement signal $\Delta C^{l-1}(s)$ is found during the analysis filtering by computing incremental coefficient δc_k^{l-1}. The update of the average signal is given over Ω_{In} and Ω_{Out} by:

$$\begin{aligned} \overline{C}^0(s) &= 0 \\ \overline{C}^l(s) &= \overline{C}^{l-1}(s) + \Delta C^{l-1}(s) \end{aligned} \tag{4}$$

and we define the residue over Ω_{In}:

$$\begin{aligned} Res^0(t,s) &= C(t,s) \\ Res^l(t,s) &= C(t,s) - \overline{C}^{l-1}(s) \end{aligned} \tag{5}$$

The analysis filtering aim is to find the refinement signal $\Delta C^l(s)$ by computing incremental coefficients δc_k^l that best represent the residue $Res^{l-1}(t,s)$ of level $l-1$.

Thus, over Ω_{In} we minimize the difference between the original signal $C(t,s)$ and the average signal $\overline{C}^{l-1}(t,s)$ of level $l-1$. Over Ω_{Out} we insuring a smooth extension of the residue by introducing a penalizing term weighted by λ. The incremental coefficients δc_k^l are given by minimizing over the group of T frame :

$$\min_{\{\delta c_k^l\}} \left(\sum_{t=1}^{t=T} \left[\sum_{s \in \Omega_{In}(t)} \varepsilon(t,s)^2 + \sum_{s \in \Omega_{Out}(t)} \lambda \|Res^l(t,s)\|^2 \right] \right) \tag{6}$$

with

$$\varepsilon(t,s) = \|\Delta C^{l-1} - Res^{l-1}(t,s)\|$$

Once the extended signal $\overline{C^L}$ is found, group of contours are easily padded (see Fig.7).

(a) Before padding (b) After padding

Fig. 7. Spatio-temporal padding illustration

3. SPATIO-TEMPORAL CODING OF OBJECT SHAPE

3.1. IPB scheme

Once contours have been aligned and padded, we may represent the evolution of contour positions in function of s (the universal abscissa) and t (the time) by two spatio-temporal planes (see Fig.8), as in [5].

Two parameterization methods are compared to encode this two spatio-temporal 2D planes. The first one is based on B-splines. The second method is based on a wavelet decomposition. Further, contours will be coded using an IPB scheme e.g. first contour will be coded intra (I) and others will be coded using a simple prediction (P) or a bidirectional prediction (B). Only one B frame is inserted between two successive I or P frames. Coefficients are coded with a bit plan arithmetic coder. Lossy quantization is obtained by choosing the encoded number of bit plans.

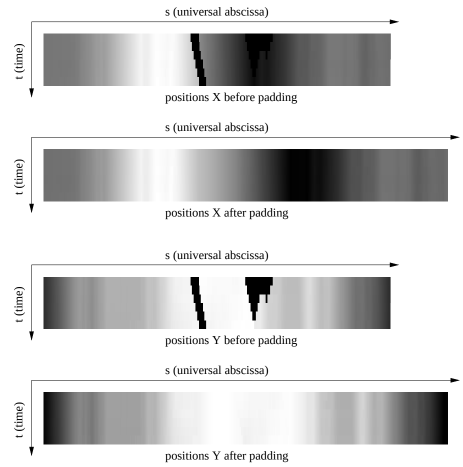


Fig. 8. Spatio-temporal plane for position X and Y. The black areas represent the contour's break

3.2. B-spline representation

A contour may be expressed as:

$$B(s) = \sum_{k=1}^{k=K} \phi(s - s_k).P_k \tag{7}$$

where P_k are controls points, ϕ are kernels of a bicubic B-spline, s is the universal abscissa and s_k is the P_k's corresponding universal abscissa.

Controls points are computed in order to best fit to original contour. It is performed thanks to a classical minimization approach:

$$\min_{P_k} \sum_{i=1}^{i=I} \|B(s_i) - C(s)\|^2 \qquad (8)$$

This minimization leads to a linear sparse system that can be easily solved using classical linear algebra tools (e.g. Conjugate Gradient).

3.3. Wavelet representation

In order to have a hierarchical representation of the group of contour and to provide scalability, dyadic wavelet decomposition is also proposed. Before performing the wavelet transformation we re-sample the group of contours (re-sample the spatio-temporal planes) in order to get a length equal to a multiple of a power of two. This enables successive 1D circular wavelet decompositions as long as there is just one coefficient left. For this decomposition we use 9/7 Daubechies' wavelet filters [8].

4. RESULTS

This section compares three methods of lossy contour coding. The MPEG4 CAE approach, and our IPB scheme with B-spline or wavelet transformation. The rate chosen over a group of frame is the average bit-rate number per contour's element. The distortion chosen for a contour is:

$$d_n = \frac{\text{number of mismatch pixels}}{\text{number of total original mask pixels}}$$

and the distortion is averaged over a group of frames.

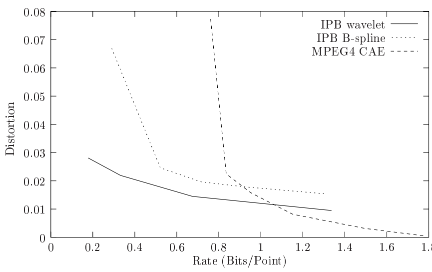


Fig. 9. Distortion vs rate (bits per contour's element) for sequence Foreman

Figure 9 shows the distortion versus the bits rate per contour's element for the Foreman sequence. First we should note that results are interesting under 1.2 bits/contour's element e.g. for lossy coding. Indeed, a Freeman lossless coding approach would perform this rate.

Using an IPB B-spline approach or IPB wavelet approach it is possible to get very low rates such as 0.4 bits/contour's element whereas MPEG4 CAE is lower bounded around 0.8 bits/ contour's element. Moreover as shown in fig.10, at low bit rates, around 0.7 bits/contour's element, quality is still acceptable which is not the case for MPEG4 CAE.

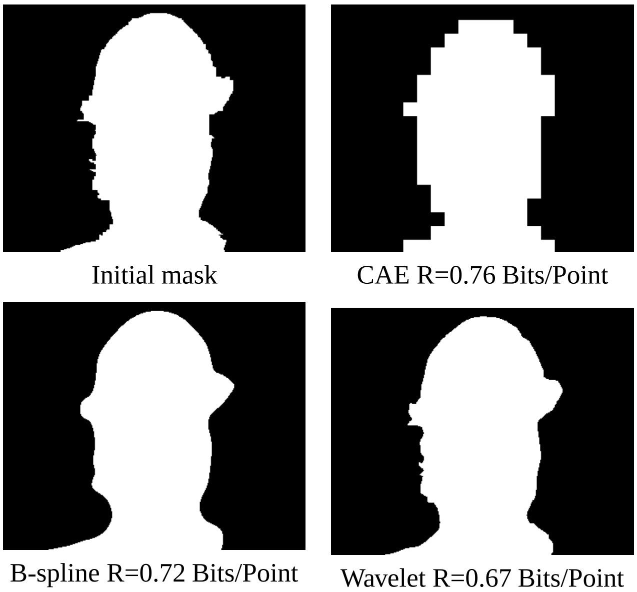


Fig. 10. Coding comparison for a rate around 0.7 bits per contour's element

Sequence	MPEG4 CAE	Wavelet	Bits saving
children	D=0.058 412 bits/fr	D=0.054 318 bits/fr	30%
Coastguard (big boat)	D=0.066 302 bits/fr	D=0.064 153 bits/fr	49%
Foreman	D=0.022 673 bits/fr	D=0.022 267 Bits/fr	60%

Table 1. Benefit of the IPB wavelet approach over MPEG4 CAE

Further the wavelet approach gives better results than the B-spline one. This may be explained by the fact that spatio-temporal de-correlation is more efficient with wavelet. Moreover, the wavelet coefficients quantization acts as choosing the more representative coefficients whereas it is difficult to choose the best quantization with the right number of controls points for B-spline.

Table 1 illustrates bits saving at low bit rate using our scheme instead of MPEG4 CAE. 30% to 60% bits saving can be obtained for an acceptable distortion level.

5. CONCLUSION

In this paper, we proposed a new efficient contour coding technique. This technique takes benefit of motion compensation, improved mapping, contours alignment and spatio-temporal padding. Thus, the evolution of the contour trough time and space is very smooth. Using wavelet coding and an IPB scheme we performs better results than MPEG4 CAE at low bit rates and thanks to intrinsic smooth spatio temporal properties the visual aspect is acceptable. Future work will deal with the integration of our shape coder in a fully optimized object based video coder.

6. REFERENCES

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

[2] C. Le Buhan Jordan, F. Bossan, and T. Ebrahimi, "Scalable shape representation for content-based visual data compression," *ICIP International Conference on Image Processing*, vol. 1, pp. 512–515, Oct. 1997.

[3] A. Katsaggelos, L. Kondi, F. Meier, J. Ostermann, and G. Schuster, "Mpeg-4 and rate-distortion-based shape-coding techniques," *IEEE Proc., special issue on Multimedia Signal Processing*, vol. 86, no. 6, pp. 1126–1154, June 1998.

[4] N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2d shape sequence," *ICIP International Conference on Image Processing*, vol. 1, pp. 29–32, Oct. 1997.

[5] T. Yoshida, T. Asami, and Y. Sakai, "Compression of moving contours on spatio-temporal 2-d plane," *ICIP International Conference on Image Processing*, vol. 1, pp. 276–280, Oct. 1998.

[6] L. Bonnaud and C. Labit, "Multiple occluding objects tracking using a non-redundant boundary-based representation for image sequence interpolation after decoding," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 426–429, Oct. 1997.

[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] 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.