

B-Spline Active Contour with Handling of Topology Changes for Fast Video Segmentation

Frederic Precioso

*I3S Laboratoire, UPRES-A 6070 CNRS, Université de Nice-Sophia Antipolis, 2000 route des Lucioles,
 F-06903 Sophia-Antipolis, France
 Email: precioso@i3s.unice.fr*

Michel Barlaud

*I3S Laboratoire, UPRES-A 6070 CNRS, Université de Nice-Sophia Antipolis, 2000 route des Lucioles,
 F-06903 Sophia-Antipolis, France
 Email: barlaud@i3s.unice.fr*

Received 26 July 2001 and in revised form 13 February 2002

This paper deals with video segmentation for MPEG-4 and MPEG-7 applications. Region-based active contour is a powerful technique for segmentation. However most of these methods are implemented using level sets. Although level-set methods provide accurate segmentation, they suffer from large computational cost. We propose to use a regular B-spline parametric method to provide a fast and accurate segmentation. Our B-spline interpolation is based on a fixed number of points 2^j depending on the level of the desired details. Through this spatial multiresolution approach, the computational cost of the segmentation is reduced. We introduce a length penalty. This results in improving both smoothness and accuracy. Then we show some experiments on real-video sequences.

Keywords and phrases: video segmentation, region-based active contour, parametric active contour, B-spline, interpolation.

1. A REGION-BASED ACTIVE CONTOUR

Segmentation of moving objects in video sequences is a real challenge to video processing specially to broadcasting and indexing applications for MPEG-4 and MPEG-7.

Active contours methods are efficient solutions for segmentation since they involve intrinsic geometry. Active contour methods are based on the evolution of a contour minimizing an energy criterion. Contour-based active contours were first introduced in [1] and were implemented using parametric methods. In the earliest contour-based methods, the image features defining the energy deal only with contour terms (image gradient) [2].

More recently region-based active contours were introduced in [3, 4, 5, 6]. These region-based active contours include terms describing the different regions such as statistical spatial features (mean, variance, ...) or temporal gradient.

A video sequence can be considered as a scene with a background (the motionless region) and a foreground (the moving objects). In this case, regions-based informations are represented by functions depending on the temporal gradient so-called descriptors.

The general problem for video segmentation is to compute, for each frame n , the regions minimizing the following criterion:

$$J(\Omega_{n,\text{in}}, \Omega_{n,\text{out}}, \Gamma_n) = \int_{\Omega_{n,\text{out}}} k^{(\text{out})}(x, y) dx dy + \int_{\Omega_{n,\text{in}}} k^{(\text{in})}(x, y) dx dy + \int_{\Gamma_n} k^{(b)}(x, y) ds. \quad (1)$$

For each frame n , Γ_n is the frontier between the domains, $k^{(b)}$ is a descriptor of the contour Γ_n , $k^{(\text{out})}$ is the descriptor of the background domain $\Omega_{n,\text{out}}$ and $k^{(\text{in})}$ is the descriptor of the object domain $\Omega_{n,\text{in}}$ (Figure 1).

In order to solve this problem, Jehan-Besson et al. [6] introduced a dynamic region-based active contour segmentation. In their method, the active contour evolves from an initial position $\Gamma_{n,0}$ towards the boundary of the objects according to the minimization of the general energy criterion

$$J_n(\tau) = \int_{\Omega_{n,\text{out}}(\tau)} k^{(\text{out})} d\sigma + \int_{\Omega_{n,\text{in}}(\tau)} k^{(\text{in})} d\sigma + \int_{\Gamma_n(\tau)} k^{(b)} ds. \quad (2)$$

In this expression, τ is the parameter for the evolution of the contour in the current frame n ; $k^{(\text{out})}$, $k^{(\text{in})}$, and $k^{(b)}$ are descriptors as above. Domains depend on the parameter τ : $\Omega_{n,\text{out}}(\tau)$ is the new background domain, $\Omega_{n,\text{in}}(\tau)$ is the new

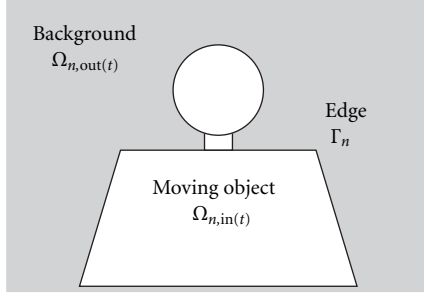


FIGURE 1: Domains definition.

object domain, and $\Gamma_n(\tau)$ is the frontier between these domains.

Then, this criterion is differentiated in an Eulerian framework [7]:

$$J'(\tau) = \int_{\Gamma(\tau)} (k^{(out)} - k^{(in)} - k^{(b)} \cdot \kappa + \nabla k^{(b)} \cdot \vec{N}) \cdot \vec{v} ds. \quad (3)$$

Thus, according to the inequality of Cauchy-Schwartz, the fastest decrease of $J(\tau)$ is obtained by choosing $\vec{v} = \nu \vec{N}$. Thus we obtain the following partial differential equation (PDE):

$$\frac{\partial \Gamma_n(t)}{\partial t} = \nu \vec{N} = \vec{F}, \quad \Gamma_n(0) = \Gamma_{n,0}. \quad (4)$$

From the initial curve $\Gamma_n(0)$, the contour $\Gamma_n(t)$ evolves following its normal direction \vec{N} with the force definition

$$F = k^{(in)} - k^{(out)} + k^{(b)} \kappa, \quad (5)$$

where κ is the contour curvature.

For *video* segmentation, we use the following descriptors: $k^{(out)}$ is the temporal gradient $|S_n - S_{n-1}|$, where S_n is the frame n in the sequence, $k^{(in)}$ is a constant called α_c , and $k^{(b)}$ is a constant called λ .

Thus, the expression of the force of evolution is given by

$$F = |S_n - S_{n-1}| - \alpha_c + k^{(b)} \kappa. \quad (6)$$

Different implementations can be used to compute active contour method evolution.

- *Nonparametric implementation*: level-set technique is a relevant solution. An efficient method has been proposed to implement region-based video segmentation [6]. Its main advantage is the segmentation accuracy. Furthermore, it provides an implicit management of topological changes. On the other hand, the computational cost is important.

- *Parametric implementations*: the B-spline interpolation of the contour presented in this paper highly reduces the computational cost while topological changes are not easily managed.

In this paper, we propose a fast parametric method using B-spline interpolated active contour. Although standard B-spline interpolation method has already been applied in the framework of contour-based active contours, the basic idea

of our method is to use *regular B-spline interpolation with length constraint* for region-based video segmentation. This results in accurate region-based segmentation and fast multiresolution parametric computation.

2. B-SPLINE CURVES INTERPOLATION

Among many interpolations of curves, B-splines provide a C^2 regularity in each point of the curve, even in the interpolated points. Moreover, B-splines have been shown to be efficient approaches to interpolate curves [8, 9]. Thus B-splines appear as the best tradeoff between accuracy and computational cost (number of points).

A cubic B-spline curve is given by a polynomial expression as (see [10])

$$S_i(s) = Q_{i-1}B_{S_{i-3}}^4(s) + Q_iB_{S_{i-2}}^4(s) + Q_{i+1}B_{S_{i-1}}^4(s) + Q_{i+2}B_{S_i}^4(s), \quad (7)$$

where s is the parameter of the curve (in fact the curvilinear abscissa), the points Q_i are the control points of the B-spline (they are virtual points), and $B_{S_i}^4(s)$ are polynomial expressions (with 2 components) defining basic functions of the B-spline.

In all these expressions Q and P represent points in a frame. Hence they both have 2 components as with $S_i(s)$ which is the parametric equation of the arc between P_i and P_{i+1} . Thus $S_i(s)$ represents $(x_i(s), y_i(s))$ with $P_i = (x_i(s_i), y_i(s_i))$ and $P_{i+1} = (x_i(s_{i+1}), y_i(s_{i+1}))$.

If all couples (P_i, P_{i+1}) are evenly sampled along the curve, for all $i \in [0, \dots, n-2]$, $\Delta s_i = \|s_{i+1} - s_i\|$, a new parameterization for each arc into $[0, 1]$ leads to an easier computation. Hence t is the new parameter, with $t \in [0, 1]$. Basics functions $B_{S_i}^4(t)$ are identical for all arcs. The curves of the interpolation become *uniform* B-splines. The arc equation is (see [10])

$$S_i(t) = \left(-\frac{1}{6}Q_{i-1} + \frac{1}{2}Q_i - \frac{1}{2}Q_{i+1} + \frac{1}{6}Q_{i+2} \right) t^3 + \left(\frac{1}{2}Q_{i-1} - Q_i + \frac{1}{2}Q_{i+1} \right) t^2 + \left(-\frac{1}{2}Q_{i-1} + \frac{1}{2}Q_{i+1} \right) t + \frac{1}{6}Q_{i-1} + \frac{2}{3}Q_i + \frac{1}{6}Q_{i+1}. \quad (8)$$

The computational cost of a regular sampling is lower than handling a specific equation for each segment of B-spline curve S_i .

Each interpolated point P_i corresponds to the polynomial expression $S_i(t)$ value when $t = 0$. From expression (8) we obtain the relation between n interpolated points P_i , $i \in [0, \dots, n-1]$, and $n+2$ control points, Q_i , $i \in [-1, \dots, n]$,

$$S_i(0) = P_i = \frac{1}{6}(Q_{i-1} + 4Q_i + Q_{i+1}). \quad (9)$$

We have to define the relation between the n interpolated points P_i and the $n + 2$ control points Q_i . In order to close the contour, the first and the last point Q_0 and Q_{n-1} are used twice. Thus

$$Q_{-1} = Q_{n-1}, \quad Q_n = Q_0. \quad (10)$$

Hence the number of control points Q needed to interpolate the n points P is n . The circulating $n \times n$ system defining the relation between P and Q is deduced from (9) and (10) as follows:

$$\begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{pmatrix} = \begin{pmatrix} \frac{4}{6} & \frac{1}{6} & 0 & 0 & \frac{1}{6} \\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{4}{6} \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{n-2} \\ Q_{n-1} \end{pmatrix}. \quad (11)$$

We compute as many control points Q as interpolated points P .

The computation of the coefficients of polynomial components of each B-spline segment $S_i(t)$ is straightly deduced from the control points Q (9).

Thus, curvature $\kappa_i(t)$ is computed at each interpolated point P_i through the relation

$$\kappa_i = \kappa_i(0) \quad \text{with} \quad \kappa_i(t) = \frac{x'_i(t)y''_i(t) - x''_i(t)y'_i(t)}{(x'_i(t)^2 + y'_i(t)^2)^{3/2}}, \quad (12)$$

where $x_i(t)$ and $y_i(t)$ are the first and the second components of $S_i(t)$, respectively.

The analytic components of $S_i(t)$ provide a straight computation of the normal vector at P_i

$$\vec{N}_i = \vec{N}_i(0), \quad \vec{N}_i(t) = \begin{pmatrix} \frac{-y'_i(t)}{\sqrt{x'_i(t)^2 + y'_i(t)^2}} \\ \frac{x'_i(t)}{\sqrt{x'_i(t)^2 + y'_i(t)^2}} \end{pmatrix}. \quad (13)$$

The main advantage of using C^2 B-splines interpolation is that computation of the velocity requires the evaluation of analytic expressions (12) and (13).

3. REGULAR B-SPLINE

Our algorithm adds multiresolution properties and local smoothness to the following intrinsic properties of B-spline interpolation:

- the number of points to be processed is highly reduced and fully controlled. For each resolution j we have $n = 2^j$ points;
- a local regularizing term over the smoothness of the

contour is added to the intrinsic C^2 regularity of the B-splines.

3.1. Spatial segmentation

First, we fix the number of the points of interpolation. This number will be 2^j , where j represents the step in the multiresolution process. Thus the segmentation speed is directly linked to the level of the desired details. In addition, solving system (11) can be avoided. Indeed, the relation is based on the resolution of a matrix which has a fixed size $2^j \times 2^j$ depending only on j , the multiresolution step of the spatial segmentation. Thus the inverse matrices can be computed off-line before computing the segmentation for each level of details j .

3.2. Spatial regularization

The second idea deals with the smoothness of the contour. The C^2 regularity of the B-splines curves brings an intrinsic smoothness. However, this regularization is distributed globally along the contour regardless of local variations of the curvature. The regularity of the contour is improved by introducing the penalty on the length of the contour (the third term of the energy criterion (2)).

4. HANDLING TOPOLOGY CHANGES

One of the most important drawbacks to use a parametric active contours method deals with handling topology changes.

In order to segment many objects in a frame of a video sequence, topological changes have to be completely managed. We use in this case the variation diminishing property [11] of the Bezier curves: the curve is not intersected by any straight line oftener than the interpolation points polygon. Hence in order to detect self-intersections we can consider only this polygon for a faster computation. When a contour overlapping occurs, we split the contour. Therefore, handling topology changes with this approach increases the computational cost of order N_p^2 , where N_p is the number of interpolation points. More sophisticated algorithms give an order of $N_p \log(N_p)$. Figure 2 shows the contour initialization for segmentation of two children in a video sequence. Figure 3 shows the spline contour before topological changing. Figure 4 is the result of the convergence of our algorithm with topological splitting.

5. PROPAGATION METHOD

5.1. Principle of the method

The principle is based on applying the velocity (6), over few contour points (the interpolated points). Then the evolution of the contour depends only on the evolution of these points. The C^2 continuity of the B-splines is preserved from step to step. The regularization term over the contour length depending on the curvature κ (the third term of velocity F (6)) is defined in each point and provides a local regularization. Thus the segmentation is computed using 32, then 64, and finally 128 points of interpolation through iterations of the three following processes:



FIGURE 2: Interpolating spline initialization.



FIGURE 3: Spline before changing topology.



FIGURE 4: Spline convergence after changing topology.

- (1) active contour interpolation;
- (2) descriptors computation and propagation (4);
- (3) resampling.

5.2. Detailed algorithm

Step 1 (Interpolation for resolution j)

- (a) Initialization of the contour:
 - (i) an initial contour: an ellipse, a circle, a square on the boundaries of the image, or from the analysis of a hand-made mask of segmentation;
 - (ii) a segmentation: for example if the final segmentation of the frame n is used as initialization of the frame $n + 1$;
 - (iii) a contour from a lower level of resolution (from the step $j - 1$).
- (b) From the 2^j points of interpolation P_{2^j} evenly sam-

TABLE 1: Comparison between polygon versus spline.

	κ	\vec{N}	Total
Spline	8 op./p.	12 op./p.	20 op./p.
Polygon	13 op./p.	28 op./p.	41 op./p.

pled, the 2^j virtual control points Q_{2^j} are computed (using (9)).

- (c) Points Q give the coefficients of the parametric components for each B-spline segments S_i (8).

Step 2 (Descriptors computation and propagation)

- Analytic expressions of each S_i curve give the exact value of the curvature (12), and of the normal vector (13) at each point of interpolation P_i . The temporal gradient is evaluated only in a neighborhood of the interpolated points P_i .
- Velocity (6) is computed *only along the contour* and the points P are moved to their new position following the normal direction of the contour with the velocity as range.
- Contour overlapping checking (see Section 4).
- If the average position of each point of interpolation is lower than a given threshold, we consider the segmentation achieved. *Otherwise go to Step 1(b)*.

Step 3 (Regular resampling). Resampling depends on the resolution j :

- If the highest level of resolution is not reached, the number of points of interpolation P is increased from 2^j to 2^{j+1} . Then points P are resampled evenly along the contour in order to hold the uniformity sampling of the B-splines. *Go to Step 1(a)(iii)*.
- If the highest level of resolution is reached, a new frame is considered. The number of points of interpolation is reinitialized. *Go to Step 1(a)(ii)*.

6. EXPERIMENTS

6.1. Comparison between polygon and B-splines

For our algorithm, we compute the computational cost of the curvature and the normal vector. Thanks to the C^2 continuity of B-splines, this computation comes from the evaluation of analytic expressions (12) and (13). This calculus is independent of the considered point since the B-splines are C^2 regular in each point of the contour.

Polygon-based methods use as approximation for the curvature at P_i , the length of the median in the triangle (P_{i-1}, P_i, P_{i+1}) [11]. The normal vector is defined as the average vector \vec{N} between the normal vectors \vec{N}_1 and \vec{N}_2 of previous and next polygon segment on both sides of the point P_i [12].

The computation cost comparison is given in Table 1.

6.2. Length versus smoothness

The property of cubic B-splines C^2 continuity enforces the global contour regularity. We add a local regularization over

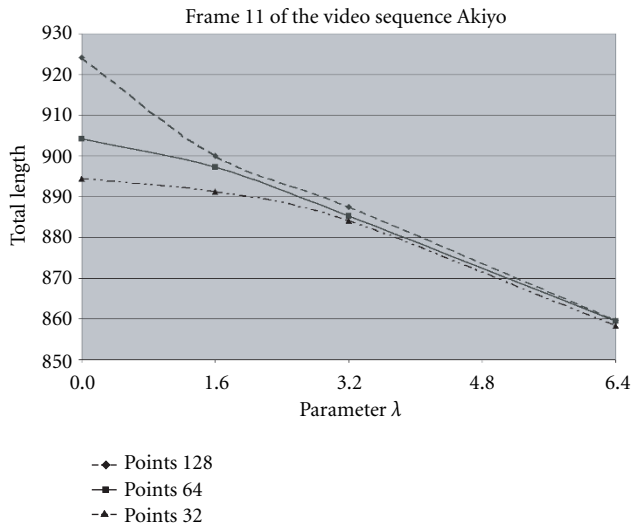


FIGURE 5: Average curve length over 50 frames (Akiyo sequence).

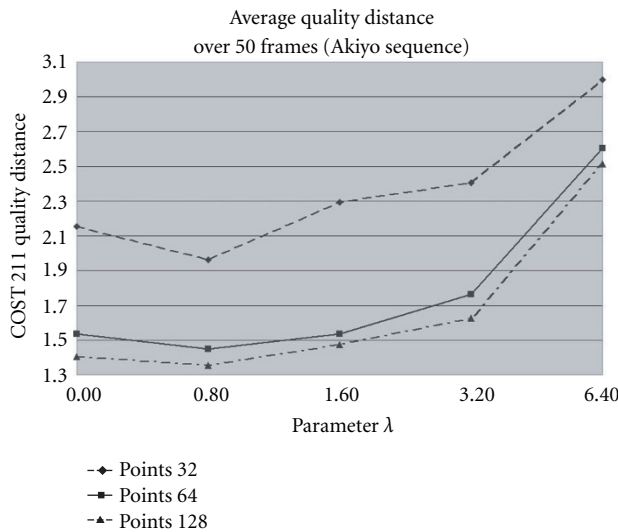


FIGURE 6: Accuracy versus smoothness.

the contour length to the intrinsic C^2 regularity of B-spline curves. With this penalty, the smoothness of the contour is locally handled at each interpolated point P . Indeed, Figure 5 shows that without this penalty ($\lambda = 0$), the length of the contour is only linked to the number of points of interpolation. Whereas when using the penalty ($\lambda \neq 0$), the length of the contour is independent of the resolution. Thus, we can use a high number of points of interpolation to improve the accuracy while maintaining the regularity of the contour.

6.3. Accuracy versus smoothness

Introducing the penalty on the length of the contour avoids classical drawbacks of splines interpolation methods such as the flickering. The accuracy is represented by the number of misclassified pixels comparing our segmentation masks to

hand-made segmentation masks. The comparison is based on the COST 211 analysis model [13]. Figure 6 shows the existence of a nonzero value for the parameter λ that gives the highest accuracy. Hence the accuracy is improved by introducing the regularization term in criterion (2).

6.4. Accuracy versus resolution

Figure 6 shows the relation between the resolution and the accuracy of the segmentation. Hence the tradeoff between accuracy of the segmentation, which requires a high number of points of interpolation, and the speed of the process has to be optimized.

7. RESULTS

We show some results on real-video sequences: Akiyo and Children (Figures 7 and 8).



FIGURE 7: Segmentation result on Akiyo sequence.



FIGURE 8: Segmentation result on Children sequence after automatic splitting.

8. CONCLUSION

In this paper, we propose a new active contour method using a regular B-spline interpolation to implement a region-based active contour segmentation algorithm.

Model deformations are controlled by criterion (2), including region-based features and a curvature term. We introduce a fixed multiresolution process and a smoothness penalty on the contour. This results in reducing highly the computational cost and improving the local regularity of the curve regardless of the resolution of the contour.

Finally, experimental results show that our regular B-spline region-based active contour method provides a fast and efficient video segmentation.

REFERENCES

- [1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [2] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [3] S. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.
- [4] A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Trans. Medical Imaging*, vol. 15, no. 6, pp. 859–870, 1996.
- [5] N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," in *Proc. International Conf. on Computer Vision*, pp. 688–694, Corfu, Greece, September 1999.
- [6] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Video object segmentation using Eulerian region-based active contours," in *Proc. International Conf. on Computer Vision*, Vancouver, Canada, July 2001.
- [7] S. Jehan-Besson, M. Barlaud, and G. Aubert, "DREAM2S: deformable regions driven by an Eulerian accurate minimization method for image and video segmentation," in *ECCV*, Copenhagen, Denmark, May 2002.
- [8] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Medical Imaging*, vol. 19, no. 7, pp. 739–758, 2000.
- [9] J. Kybic, P. Thevenaz, and M. Unser, "Multiresolution spline warping for EPI registration," in *Proc. SPIE, Wavelet Applications in Signal and Image Processing VII*, vol. 3813, pp. 571–579, Denver, Colo, USA, July 1999.
- [10] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan-Kaufmann, Los Altos, Calif, USA, 1987.
- [11] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press, Boston, Mass, USA, 4th edition, 1996.
- [12] M. Maziere and F. Chassaing, "Segmentation and tracking of video objects: suited to content-based video indexing, interactive television and production systems," in *Int. Conference on Image Processing*, Vancouver, British Columbia, Canada, 2000.
- [13] R. Mech, Ed., *Description of COST 211 Analysis Model*, COST 211^{quat} Simulation Group, Dublin, Ireland, 1998.

Michel Barlaud received his "Thèse d'État" from the University of Paris XII. He is currently a Professor of image processing at the University of Nice-Sophia Antipolis, and the leader of the image processing group of I3S. His research topics are image and video coding using scan-based wavelet transform, inverse problem using half quadratic regularization, and image and video segmentation using region-based active contours and PDEs. He is a regular reviewer for several journals, a member of the technical committees of several scientific conferences. He leads several national research and development projects with French industries, and participates in several international academic collaborations (Universities of Maryland, Stanford, Louvain La Neuve, and Naples). He is the author of a large number of publications in the area of image and video processing, and the editor of the book *Wavelets and Image Communication*, Elsevier, 1994.



Frederic Precioso was born in 1974. He received his M.S. degree in image processing and computer science from University of Nice-Sophia Antipolis in 2000. He is a Ph.D. student at the Laboratory I3S, University of Nice-Sophia Antipolis, France. His research interests deal with image and video processing, and computer vision. He is currently working on parametric active contour methods for video segmentation.

