B-SPLINE ACTIVE CONTOUR with HANDLING of TOPOLOGY CHANGES FOR FAST VIDEO SEGMENTATION

F. Precioso and M. Barlaud
I3S laboratory - UPRES-A 6070 CNRS
Université de Nice - Sophia Antipolis
2000 route des Lucioles - F-06903 Sophia-Antipolis FRANCE
precioso@i3s.unice.fr - barlaud@i3s.unice.fr

ABSTRACT

This paper deals with video segmentation for MPEG-4 and MPEG-7 applications. Region-based active contours 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.
In this paper 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 details desired. Through this spatial multi-resolution 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.

1 INTRODUCTION: A Region-Based Active Contour

Segmentation of moving objects in video sequences is a real challenge of video processing specially for 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 by [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 by [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.

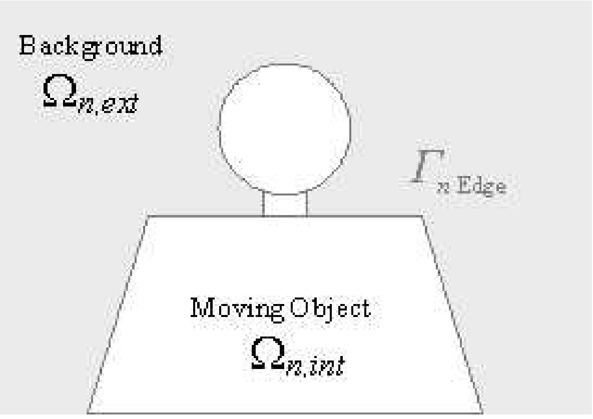

Figure 1: Domains definition

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:

$$J(\Omega_{n,in}, \Omega_{n,out}, \Gamma_n) = \int_{\Omega_{n,out}} k^{(out)}(x,y)dxdy + \int_{\Omega_{n,in}} k^{(in)}(x,y)dxdy + \int_{\Gamma_n} k^{(b)}(x,y)ds \qquad (1)$$

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

In order to solve this problem, Jehan & al [6] introduce 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:

$$J_n(\tau) = \int\limits_{\Omega_{n,out}(\tau)} k^{(out)} d\sigma + \int\limits_{\Omega_{n,in}(\tau)} k^{(in)} d\sigma + \int\limits_{\Gamma_n(\tau)} k^{(b)} ds \tag{2}$$

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

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

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

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

$$\begin{cases} \frac{\partial \Gamma_n(t)}{\partial t} & = & v\vec{N} & = & \vec{F} \\ \Gamma_n(0) & = & \Gamma_{n,0} \end{cases} \tag{4}$$

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

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

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:

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

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

- Non-parametric 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 while topology 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 multi-resolution 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 trade-off between accuracy and computational cost (number of points).

A cubic B-Spline curve is given by a polynomial expression as below [10]:

$$\begin{aligned} S_i(s) &= Q_{i-1} B_{S^4_{i-3}}(s) + Q_i B_{S^4_{i-2}}(s) \\ &+ Q_{i+1} B_{S^4_{i-1}}(s) + Q_{i+2} B_{S^4_i}(s) \end{aligned} \tag{7}$$

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 $B_{S^4_i}(s)$ 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, $\forall i \in [0, .., n-2]$ $\triangle s_i = \|s_{i+1} - s_i\|$, 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 $B_{S^4_i}(t)$ are identical for all arcs. The curves of the interpolation become *Uniform* B-Splines. The arc equation is [10]:

$$\begin{aligned} S_i(t) = &\left(-\tfrac{1}{6}\ Q_{i-1} + \tfrac{1}{2}\ Q_i - \tfrac{1}{2}\ Q_{i+1} + \tfrac{1}{6}\ Q_{i+2}\right) t^3 \\ &+ \left(\tfrac{1}{2}\ Q_{i-1} - Q_i + \tfrac{1}{2}\ Q_{i+1}\right) t^2 \\ &+ \left(-\tfrac{1}{2}\ Q_{i-1} + \tfrac{1}{2}\ Q_{i+1}\right) t \\ &+ \tfrac{1}{6}\ Q_{i-1} + \tfrac{2}{3}\ Q_i + \tfrac{1}{6}\ Q_{i+1} \end{aligned} \tag{8}$$

The computation 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 the expression (Eq. 8) we obtain the relation between n interpolated points $P_i, i \in [0, ..., n-1]$ and $n+2$ control points, $Q_i, i \in [-1, ..., n]$:

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

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:

$$Q_{-1} = Q_{n-1}, \qquad Q_n = Q_0 \tag{10}$$

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 the equations (Eq. 9) and (Eq. 10) as follow:

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

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 (Eq. 9).

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

$$\kappa_i = \kappa_i(0), \ with \ \ \kappa_i(t) = \frac{x_i'(t)y_i''(t) - x_i''(t)y_i'(t)}{\left(x_i'(t)^2 + y_i'(t)^2\right)^{\frac{3}{2}}}. \tag{12}$$

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

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

$$\overrightarrow{N_i} = \overrightarrow{N_i(0)}, \quad \overrightarrow{N_i(t)} \begin{pmatrix} \frac{-y_i'(t)}{\sqrt{x_i'(t)^2 + y_i'(t)^2}} \\ \frac{x_i'(t)}{\sqrt{x_i'(t)^2 + y_i'(t)^2}} \end{pmatrix} \tag{13}$$

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

3 REGULAR B-SPLINE

Our algorithm adds multi-resolution 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 multi-resolution process. Thus the segmentation speed is directly linked to the level of details desired. In addition, solving the system (Eq. 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 multi-resolution 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 to local variations of the curvature. The regularity of the contour is improved by introducing the penalty over the length of the contour (the third term of the energy criterion (Eq. 2)).

4 HANDLING TOPOLOGY CHANGES

One of the most important drawback 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 more often then is 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 computation cost of order N_p^2, where N_p is the number of interpolation points. More sophisticated algorithms give an order of $N_p \ log(N_p)$. **Fig.2** shows the contour initialization for segmentation of two children in a video sequence. **Fig.3** shows the spline contour before topological changing. **Fig.4** is the result of the convergence of our algorithm with topological splitting.

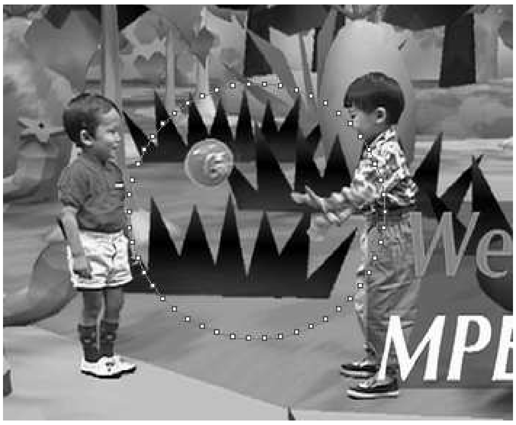

Figure 2: Interpolating Spline Initialization

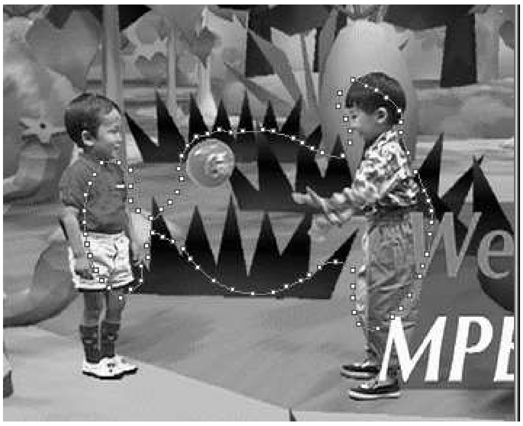

Figure 3: Spline before chnging topology

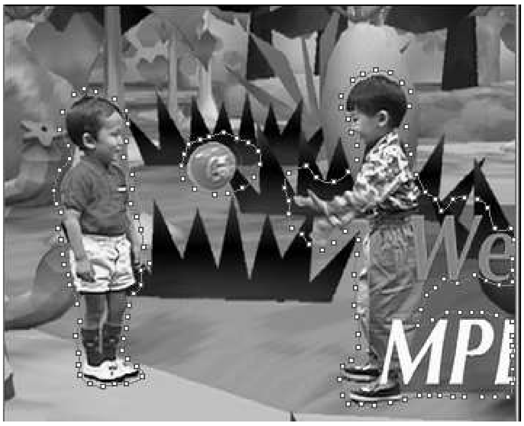

Figure 4: Spline Convergence after changing topology

5 PROPAGATION METHOD

5.1 Principle of the method

The principle is based on applying the velocity (Eq. 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 (Eq. 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:

1. Active contour interpolation
2. Descriptors computation and Propagation (Eq. 4)
3. Resampling

5.2 Detailed Algorithm

1. **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 sampled, the 2^j *virtual* control points Q_{2^j} are computed (using Eq. 9).

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

2. **Step 2: Descriptors Computation and Propagation**
   - Analytic expressions of each S_i curve give the exact value of the curvature (Eq. 12), and of the normal vector (Eq. 13) at each point of interpolation P_i. The temporal gradient is evaluated only in a neighborhood of the interpolated points P_i.
   - Velocity (Eq. 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 GOTO Step 1.(b)

3. **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.

  GOTO 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.

  GOTO 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 (Eq. 12) and (Eq. 13). This calculus is independent on the point considered 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 side of the point P_i [12].

The computation cost comparison is given in the following table (**Tab.1**).

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

Table 1: Comparison Polygon vs Spline

6.2 Length vs Smoothness

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

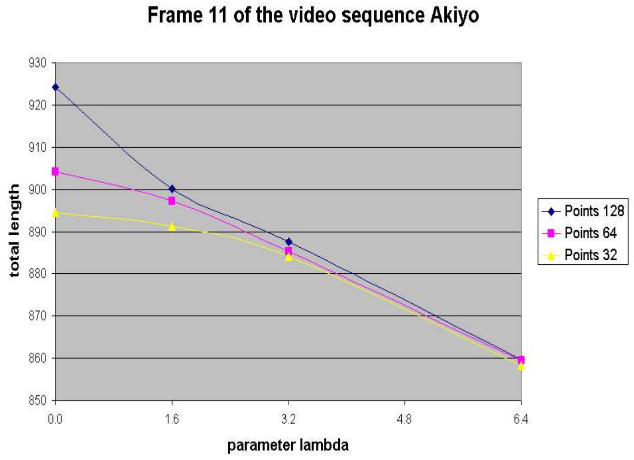

Figure 5: Length vs lambda

6.3 Length vs Smoothness

Introducing the penalty over 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]. Diagram (**Fig.6**) shows the existence of a non zero value for the parameter λ that gives the highest accuracy. Hence the accuracy is improved by introducing the regularization term in criterion (Eq. 2).

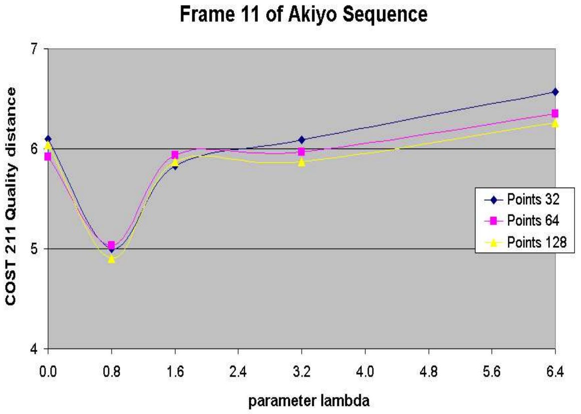

Figure 6: Accuracy vs Smoothness

6.4 Accuracy vs Resolution

The diagram (**Fig.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 (**Fig.7 and Fig.8**).

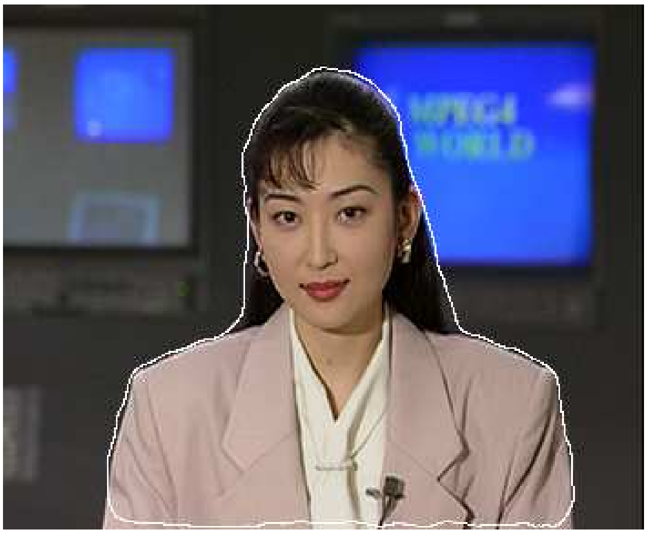

Figure 7: Segmentation Result on Akiyo Sequence

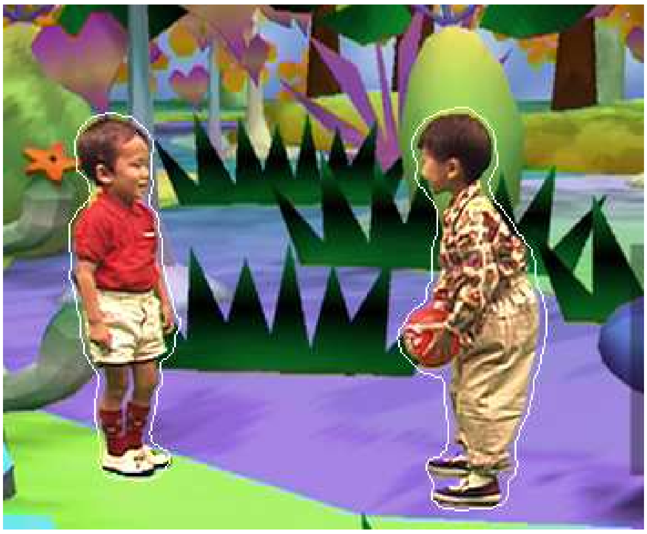

Figure 8: Segmentation Result on Children Sequence after automatic splitting

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 (Eq. 2), including region-based features and a curvature term. We introduce a fixed multi-resolution process and a smoothness penalty over the contour. This results in reducing highly the computational cost and improving the local regularity of the curve regardless to 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," *IJCV*, vol. 1, pp. 321–332, 1988.

[2] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IJCV*, 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 PAMI*, vol. 18, 1996.

[4] A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Trans. Med. Imag.*, vol. 15, 1996.

[5] N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," in *ICCV*, Corfu Greece, 1999.

[6] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Video object segmentation using eulerian region-based active contours," in *ICCV*, Vancouver, Canada, 2001.

[7] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Dreams: 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," in *IEEE Transactions on medical imaging*, july 2000, vol. 19.

[9] J. Kybic, P. Thevenaz, and M. Unser, "Multiresolution spline warping for epi registration," *SPIE*, vol. 3813, pp. 571–579, 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, Californie, 1987.

[11] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design, A Practicle Guide, 4th Edition*, Academic Press, 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 *ICIP*, Vancouver Canada, 2000.

[13] Roland Mech, *Description of COST 211 analysis model*, COST 211^{quat} simulation group, Dublin, 1998.