Proceedings of ACIVS 2002 (Advanced Concepts for Intelligent Vision Systems), Ghent, Belgium, September 9-11, 2002

AN EXTENDED SNAKE MODEL FOR REAL-TIME MULTIPLE OBJECT TRACKING

[1,2]*Sébastien Lefèvre*, [1]*Jean-Pierre Gérard*, [1]*Aurélie Piron and* [1]*Nicole Vincent*

[1]`(lefevre,vincent)@univ-tours.fr`
[1]Laboratoire d'Informatique, E3i / Université de Tours, 64 avenue Portalis, 37200 Tours - FRANCE
[2]AtosOrigin, 19 rue de la Vallée Maillard, BP1311, 41013 Blois Cedex - FRANCE

ABSTRACT

In this paper we present a snake model dedicated to real-time multiple object tracking. This model is able to track moving non-rigid objects in video sequences obtained from moving camera. Contrary to other approaches, here the snake deformation is performed without any preprocessing and the tracking method does not rely on any motion estimation step. The tracking method is based on two simple successive steps which are snake initialisation and snake deformation. Avoiding complex additional steps allows us to propose a tracking method which can be performed in real-time.

Two specific extensions of classical applications are proposed in this paper. The first one gives the snake the ability to divide in several contours if necessary. It is particulary useful when two objects which were close together move apart. It also improves the robustness of the tracking method against noise characterized by high gradient values. The second extension deals with multiresolution processing of the image. In order to limit computation time, we propose to process low-resolution frames instead of original ones. A quality criteria is then used to determine if the result obtained is correct or not. In the latter case, the current frame is analysed at a higher resolution.

The proposed extended snake model has been successfully applied to real-time multiple object tracking of football players in football game video sequences. Players positions obtained are then further used in a real-time football video analysis tool.

1. INTRODUCTION

Object tracking is a crucial step in any computer vision system. Most often objects that are involved are not rigid. When dealing with non-rigid objects, it is necessary to use a more sophisticated method. Several methods have been developed to model non-rigid objects. For example they can be modelled by snakes or active contour models. Several methods based on snakes have already been proposed for object tracking but only few of them are able to track non-rigid objects in real time due to the computation complexity induced by active contour models. Besides when the source is also moving, tracking methods often include an additional motion compensation step, which is also time consuming.

We are concentrating our study on two main goals. On the one hand we want to realise a real-time process and on the other hand we want to make possible non-rigid object tracking, more precisely we are concerned about deformable objects with possible crossing trajectories. In this case, two objects can be seen as only one when an occlusion phenomenon occurs. Besides in the general case of course it is needed to process data from a possible moving camera source. In order to achieve real time processing of colour video sequences the computation complexity has to be minimised.

Among the so many methods we have excluded the level set methods for their time consuming characteristics and we have chosen to refer to active contours. Even if active contours need less computational resources than level set methods, they still often lead to time consuming methods. So many developments have to be realised in order to achieve the goals we have precised.

Indeed in our approach we have chosen not to perform any motion compensation nor any preprocessing. This distinguishes our method from most of the traditional approaches and unables to gain in computation time. But this condition in fact leads to the need of a high robustness of the methods that are involved. In our case, computation is also reduced by the use of a multiresolution scheme.

The random trajectory of the object is managed by a possible snake scission step which allows to track objects being close together and then moving apart.

First active contour models will be introduced and traditional object tracking approaches based on snakes will be reviewed. We will then recall the energies used in our snake model and the two steps composing the tracking method. We will next described the scission step which allows to extend our method to multiple object tracking. The use of a multiresolution analysis will be also presented. Finally some results on football players tracking in football game colour video sequences will be presented and commented.

2. ACTIVE CONTOURS AND OBJECT TRACKING

Snakes, or active contour models, have been introduced by Kass et al [1]. They belong to deformable template models [2] and are widely used in computer vision, mainly for image segmentation or object tracking [3]. In this section we introduce the active contour model and review some traditional object tracking approaches based on snakes.

2.1. Active Contour Model

An active contour can be represented as a parametric curve

$$v(s) = [x(s), y(s)]\,,\ s \in [0,1] \tag{1}$$

evolving through time and which can be either closed or not. The active contour evolution is performed thanks to the minimization of an energy function. This function can be defined in the continuous domain as

$$E = \int_0^1 [\alpha \cdot E_{int}(v(s)) + \beta \cdot E_{ext}(v(s))]\, ds \tag{2}$$

where E_{int} and E_{ext} represent respectively the internal and external energies. Internal energy represents physical properties of the contour whereas external energy links contour to data contained in the image (as gradient or intensity). α and β are coefficients used to give more or less influence to the different energies. Most of the time, internal and external energies may be defined as combination of other energies. Description of classically-used energies and their interpretations are given in [4].

Several approaches have been proposed for the implementation of the active contour model. We can mention well-known variational calculus [1], dynamic programming [5], or greedy algorithm [6]. In a comparative study [7], it has been shown the greedy algorithm is faster than other approaches. In this approach, the implementation of active contour model is discrete and local. Every point of the parametric curve is iteratively deformed until convergence using a given termination criterion. See work by Wong et al [8] for a critical review of existing termination criteria. For each point v_i of the active contour, the energy function is computed on every point n_i^j belonging to the neighbourhood of v_i. The point n_i with minimum energy replaces the point v_i if $E(n_i) < E(v_i)$. The energy for a point v_i is given by:

$$E(v_i) = \alpha \cdot E_{int}(v_i) + \beta \cdot E_{ext}(v_i) \tag{3}$$

So the definition in the discrete domain of the energy function of the snake is:

$$E = \sum_{i=1}^{m} E(v_i) \tag{4}$$

where m is the number of points composing the snake.

2.2. Snakes for Object Tracking

Snakes are particularly adapted to tracking of non-rigid objects. So many snake-based tracking methods have been proposed in the literature. We present some studies in next part. First approaches were proposed by Kass et al and by Leymarie and Levine [9] using the final snake in the previous frame as the initial snake in the current frame. In order to deal with important motion or low frame rate, Terzopoulos and Szeliski [10] and Peterfreund [11] propose to use a Kalman filter for motion estimation. Vieren et al [12] set a global snake on the borders of the image to detect and track new objects entering in the scene. When dealing with complex background, new energy term can be added as the texture energy introduced by Delagnes et al [13] or the motion compensation error term used by Pardas and Sayrol [14]. Lam and Yuen [15] modify the energy formulation and propose a tracking method giving better results than tracking based on greedy algorithm from [6]. Radial modelling of the contour has also been proposed by Denzler and Nieman in [16] and Chen et al in [17] to prevent crossings in the contour and to reduce the energy minimisation problem from a 2D image plane to a 1D space. Real-time tracking with snake has been reviewed by Blake and Isard in [3]. Finally several snake-based approaches for football players tracking have been proposed, including a segmentation preprocessing step in [18] or without any in [19].

3. PROPOSED ALGORITHM

We have already proposed a fast snake-based method to track football player [19]. Here we recall in this section the energies used in the model. We will also present our tracking method based on two successive steps: snake initialisation and snake deformation.

3.1. Definition of the Energies

The snake deformation is performed iteratively thanks to the minimization of an energy function. As defined in (3), the energy is composed of internal and external energies.

3.1.1. Internal Energy

The internal energy is itself composed of three energies linked to continuity, balloon, and curvature forces. The definition of the internal energy is given in:

$$E_{int} = a \cdot E_{con} + b \cdot E_{bal} + c \cdot E_{cur} \tag{5}$$

where E_{con}, E_{bal}, E_{cur} represent energies linked respectively to continuity, balloon, and curvature forces. As for (3), coefficients (here a, b, and c) are used to give more or less importance to the different energies.

Continuity force influences the curvature of the model. It forces the points of the contour to be equally distant, so the contour tends to be a circle. The energy $E_{con}(n_i^j)$ is defined for a point n_i^j in the neighbourhood of v_i as the absolute value of the difference between two distances: the average distance $\bar{n}$ between two successive points of the contour (which is computed at each iteration) and the distance from the previous contour point v_{i-1} to the point n_i^j.
Balloon force has been introduced by Cohen [20] and allows the contour to extend or to retract. We will use a reduction force to force the area of the snake to decrease. The energy $E_{bal}(n_i^j)$ is defined as the scalar product of the normal vector of the contour at the point v_i with the vector $\overrightarrow{v_i n_i^j}$. The direction of the normal vector allows the contour to grow or to reduce.
The last force used in the internal energy is the curvature. The goal of this force is to avoid isolated points which will not be coherent with the global shape. So the energy $E_{cur}(n_i^j)$ is proportional to a discrete expression of the second order derivative.

3.1.2. *External Energy*

In a similar way to the internal energy, the external energy is a linear combination of several terms. The two energies used here are linked to gradient and colour forces:

$$E_{ext} = d \cdot E_{gra} + e \cdot E_{col} \tag{6}$$

where E_{gra} and E_{col} represent energies linked respectively to gradient and colour forces.
The gradient force is based on a combination of the gradients computed on the three colour components (red, green, and blue). It makes the model to stop on the actual contours present in the image. We use Sobel operator to compute an approximation of the gradient, which is then thresholded. The use of a threshold allows us to eliminate most of false contours and so to increase robustness to noise. In order to limit processing time, the gradient is computed once per frame and only on a subpart of the image around the initial snake position.
The colour force is defined using a priori information about the colours present in the image background. More precisely, average colour of the background can be computed using some appropriate method (as [21]). For each pixel, the energy E_{col} is then defined as the difference between its colour and the average background colour. In order to limit noise, we also threshold the value obtained. By this way we force the snake to remain on the extern contour of the tracked object.

3.2. Tracking Method

The tracking method we proposed in [19] is based on two steps performed successively on every frame of the video sequence. First the snake is initialised using result from the previous frame and is next deformed using energies defined above. As it was precised previously, the tracking method does not need any preprocessing nor motion estimation / compensation step.

3.2.1. *Snake Initialisation*

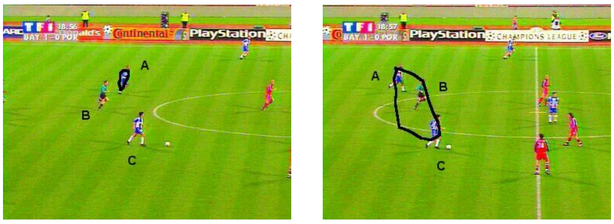
The first step consists in initialising the snake on the current frame. Once again, this is done in two steps. First a rectangle parallel to the image border is created as a bounding box around the final snake obtained from the previous frame. Next the size of the rectangle is increased in order to englobe the expected contour at the current frame. As we are using a discrete implementation of the snake model, we set the points defining the snake all along the contour of the rectangle. A specific initialisation is necessary for the first frame of a video sequence, where no a priori information is available about the contour. In this case, the initialisation can be performed either manually or automatically (using a foreground extraction method as [21]).

3.2.2. *Snake Deformation*

Once the snake has been initialised on one frame, it is then deformed and reduced in order to fit the object contour and to get the final result. The snake reduction is done until convergence using the process described previously. So we use a reducing balloon force rather than an increasing one. This can be justified by the fact that the area around the object (*i.e.* background) is more homogeneous than area inside the object which can contain high gradient values. This is often the case when the object shape is not rigid. In the opposite case, it would be better if the initialisation could be achieved using an inner contour and an expending balloon force.
From this starting point we will introduce many improvements as we will be able to track several objects at the same time in real time.

4. MULTIPLE OBJECT TRACKING

The tracking method described in the previous section is able to track moving objects in colour video sequences acquired from a moving camera. However, the tracking fails when moving objects have very close positions. Indeed, when the tracked object is close to another object, the initial snake will enclose both objects together. Once the objects move away one from the other, the snake model presented in [19] is unable to fit only one of the tracked object and continues to track both objects as only one. Another limitation of this snake model was its sensitivity to high gradient values of background pixels (*e.g.* white lines in football scenes for instance). Figure 1 illustrates these two problems.



(a) (b)

Figure 1: Illustration of incapacity of the snake to deal with close moving objects and with high gradient values of background pixels. Object tracking is correct in frame (a) and fails in frame (b) when B player comes across A player.

So we propose here an extension of the initial snake model to solve problems described above. It introduces a scission step which divides the snake in several contours if necessary and a decision step which keeps only relevant contours. The scission is based on internal snake features instead of external features as in [22]. This extension allows us also to track multiple objects at the same time. In order to limit computation time, scission and decision steps are perfomed only once per frame when the final contour has been obtained.

4.1. Scission Step

In order to divide the snake into several contours, we check the length of every segment $v_i v_{i+1}$ connecting two successive points v_i and v_{i+1} of the contour. When the length of a segment is higher than a threshold, the segment is divided each time two such segments are encountered along the snake, and several new contours are introduced that are limited to those segments (figure 2). The threshold is computed as the product of a fixed coefficient and the average distance $\bar{n}$ between two successive points of the contour. The concerned segments are divided into three equal parts. Let p_i^1 and p_i^2 denote the two intermediary points between v_i and v_{i+1}. The three segments created are $v_i p_i^1$, $p_i^1 p_i^2$, and $p_i^2 v_{i+1}$. Let us suppose the contour has to be divided at segments $v_i v_{i+1}$, $v_j v_{j+1}$, and $v_k v_{k+1}$. The snake will then be divided into three contours, which contain respectively the additional sets of segments:

$$\Omega_1 = \{v_i p_i^1,\ p_i^1 p_k^2,\ p_k^2 v_{k+1}\} \quad (7)$$

$$\Omega_2 = \{v_j p_j^1,\ p_j^1 p_i^2,\ p_i^2 v_{i+1}\} \quad (8)$$

$$\Omega_3 = \{v_k p_k^1,\ p_k^1 p_j^2,\ p_j^2 v_{j+1}\} \quad (9)$$

This process is illustrated in figure 2.

4.2. Decision Step

The scission step allows to divide a snake into several ones. It is followed by a decision step to determine the relevant

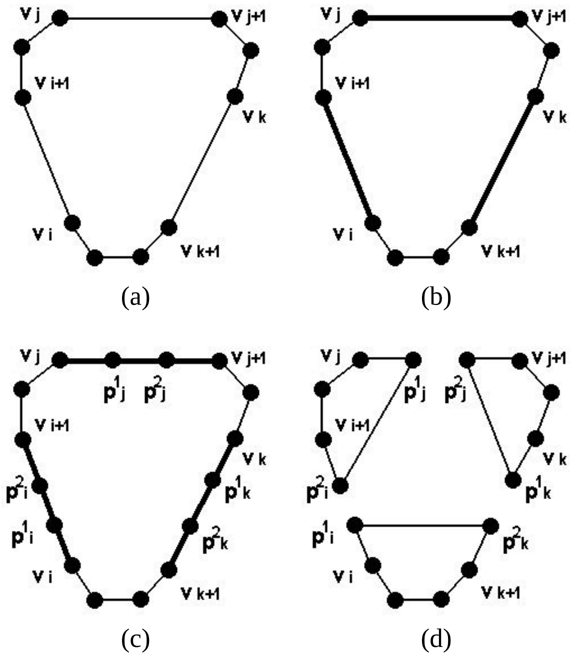


Figure 2: Successive steps in the scission process. The initial contour (a) is analysed to determine segments to be divided (b). For every concerned segment, two new points are created (c). Finally sets of segments are reorganised to build the final contours (d).

snakes from the set of contours created during the scission step. Then the coherence is verified. Indeed, the set of new contours may contain some snakes which fit noise or background pixels and which are not relevant for the tracking process. So we check the size and the shape of the new contours to discard contours consisting of a point, a line, or contours characterized by a very small or very large surface.

The other problem we want to solve is computation time that we can reduce using a multiresolution format of the images processed.

5. MULTIRESOLUTION ANALYSIS

In order to limit computation time, we propose to extend our initial snake model to a multiresolution scheme in the image representation itself. Several approaches for multiresolution modelling of active contours have already been proposed in the literature [23, 24, 25]. They consist in snake evolution on a coarse-to-fine basis. First the snake is deformed at a coarse resolution. Next the result is used as an initialisation for a snake evolution at a finer resolution. This process is iterated until the current resolution is the original one.

The multiresolution approach proposed in this paper consists in a similar coarse-to-fine analysis using a termination criterion linked to correctness of the resulting snake. So this approach is less time consuming than previously published ones. It is particularly useful when the contour obtained at coarse resolution is sufficient to process next frame. In case of a contour at fine resolution is needed, the coarse-to-fine analysis is performed until the orginal resolution without taking into account the termination criterion.

The proposed approach consists in successive analyses of one image at different resolutions. Frames are first analysed at a very low resolution. If the snake obtained after the deformation step is not correct (using the decision criteria described in previous section), the image is analysed at a higher resolution where the new width and height are twice the previous ones.

Figure 3 shows several representations of an image at different resolutions. Our algorithm is able to process images at different resolutions, from original resolution r_0 to lowest resolution r_5 where image size has been reduced by a factor of $2^{5\times2} = 1024$. On average, the minimum resolution for which correct tracking results are obtained is r_3, that is to say the image size has been reduced by a factor of $2^{3\times2} = 64$. However correct results can be obtained in some cases with lower resolution.

In order to achieve robustness of the algorithm against change in resolution, some parameters are dependant of the resolution. These parameters are the number of iterations, the number of snake points, the size of the initial rectangle contour, and the gradient threshold. On the opposite, energy coefficients and neighbourhood size do not depend on the resolution level.

resolution r_0
(size is 384×288 pixels)

resolution r_1
(size is 192×144 pixels)

resolution r_2
(size is 96×72 pixels)

resolution r_3
(size is 48×36 pixels)

resolution r_4
(size is 24×18 pixels)

resolution r_5
(size is 12×9 pixels)

Figure 3: Representation of an image at different resolutions. Resolution decreases from top left image (finest and original resolution r_0) to bottom right image (lowest resolution r_5 analysed by our algorithm). Images have been rescaled in order to see resolution differences.

6. RESULTS AND DISCUSSION

In this section we first describe the different parameters of our method and discuss about how we set them. Next we illustrate our contribution using some result images. Finally we analyse processing time of monoresolution and multiresolution methods.

6.1. Method Parameters

The method proposed here has been tested on football video sequences composed of images with size equal to 384×284 pixels and with colour coding in 24 bits. In order to limit computation time, we process pixels directly in the RGB colour space and we do not perform any conversion to another representation space. The goal is to track football players in the video sequence acquired with a moving camera.

The snake is initially composed of m points at the finest resolution. The number of points has a direct influence on the result obtained on the one hand and the processing time required on the other hand. Indeed, when the resulting shape has to be as precise as possible, the snake model will be composed of a high number of points. In this case, the final snake will better fit the tracked object. But the deformation step will also be characterized by a higher computational time. On the contrary, when the application requires only object position and when the shape precision is not of the greatest importance, the number of points can be reduced. It induces a lower processing time. We are concerned with real time tracking of moving objects where snake position is more important than precision of the snake shape. So we use a low number of points, that is to say $m = 16$.

At this resolution level, the maximum number of iterations is 30. However, most of the time, the contour converges most of the time before this number of iterations. Indeed convergence has not been obtain for only 30 % of the frames. The neighbourhood for the search of a point with minimum local energy is 5×5 pixels. In order to further decrease process-

ing time, it is also possible to limit the maximum number of iterations and to decrease the size of the neighbourhood considered for local energy computation.
The energy coefficients α, β, a, b, c, d, and e have all been set to 1. This gives satisfactory results and in the same time computational resources are limited. Table 1 sum up the different parameters which have to be set in our algorithm.

6.2. Tracking Results

Figure 4 shows an example of tracking on a 100 frames long sequence in a simple case. In this figure the tracking has been performed at the original resolution. Without performing any object motion estimation nor camera compensation, the algorithm manages to track moving object in a moving environment.
When two objects are close together, the scission step allows to divide the snake in several contours as shown in figure 5. This allows to track separately the different objects present in the scene. However, even if the scission step proposed in this paper allows the snake to correctly track close objects, the proposed method is still unable to track objects in a complex background. In this case, gradient forces will lead the contour to background pixels and the snake will not fit the tracked object.
Tracking result obtained at resolution r_4 (image size is 256 times lower than original one) are presented in figure 6. We can note the coarseness of the snake shape which is due to the low resolution level the frames are analysed at.

6.3. Analysis of Processing Times

Processing time of the proposed algorithm (without considering image reading) is about 35 milliseconds per frame on a PC laptop (Intel Pentium III with 128 MB RAM) relying on a C implementation of the proposed approach. It allows to perform real time tracking of moving objects in a moving environment using gradient and colour information.
Contrary to theoretical justifications, standard analysis is performed quite as fast as multiresolution analysis (it is only a few milliseconds slower). This is partially due to time necessary to build the hierarchical multiresolution representation of the image. This processing can be avoided when dealing with compressed video sequences (*e.g.* MPEG or MJPEG videos) where DC coefficients can be directly used (without any decompression) to get low resolution frames. In this case, full decompression of video sequences is not necessary if the algorithm converges at a low resolution. However, the additional time necessary to build the pyramidal representation of the frames is quite low (about a few milliseconds) and the main reason of the similar observed processing times for standard and multiresolution analyses is elsewhere.
Indeed observation of similar processing time for standard and multiresolution analyses is due to the fact that standard tracking algorithm has been already optimised (*e.g.* gradient is computed only once per frame and only in the neighbourhood of the initial snake) and is already characterized by low computational cost.

7. CONCLUSION

The method described in this paper is able to track non-rigid objects in colour video sequences. Processing is performed in real time even if the source camera is moving. We introduce here two extensions of our method for multiple object tracking and multiresolution analysis. Tests on football video sequences have been presented to show the quality of the proposed method.
Future work will include use of more robust colour energies as those proposed by Gevers et al in [26] and by Ngoi et Jia in [27]. A background/foreground separation step may also be necessary to deal with object tracking on a complex background. Finally implementation of the proposed algorithm on a multiprocessor workstation is considered to continue decrease computation time.

8. REFERENCES

[1] M. Kass, A . Witkins, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988.

[2] A.K. Jain, Y. Zhong, and M.P. Dubuisson-Joly, “Deformable template models: A review,” *Signal Processing*, vol. 71, no. 12, pp. 109–129, December 1998.

[3] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*, Springer-Verlag, 1999.

[4] N.E. Davison, H. Eviatar, and R.L. Somorjai, “Snakes simplified,” *Pattern Recognition*, vol. 33, no. 10, pp. 1651–1664, October 2000.

[5] A. Amini, T. Weymouth, and R. Jain, “Using dynamic programming for solving variational problems in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855–867, September 1990.

[6] D.J. Williams and M. Shah, “A fast algorithm for active contours and curvature estimation,” *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 55, no. 1, pp. 14–26, January 1992.

[7] J. Denzler and H. Niemann, “Evaluating the performance of active contours models for real-time object tracking,” in *Asian Conference on Computer Vision*, Singapore, 1995, vol. 2, pp. 341–345.

[8] Y.Y. Wong, P.C. Yuen, and C.S. Tong, "Contour length termination criterion for snake model," *Pattern Recognition*, vol. 31, no. 5, pp. 597–606, May 1998.

[9] F. Leymarie and M.D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 617–634, June 1993.

[10] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds., pp. 3–20. MIT Press, 1992.

[11] N. Peterfreund, "Robust tracking of position and velocity with kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, June 1999.

[12] C. Vieren, F. Cabestaing, and J.G. Postaire, "Catching moving objects with snakes for motion tracking," *Pattern Recognition Letters*, vol. 16, no. 7, pp. 679–685, July 1995.

[13] P. Delagnes, J. Benois, and D. Barba, "Active contours approach to object tracking in image sequences with complex background," *Pattern Recognition Letters*, vol. 16, no. 2, pp. 171–178, February 1995.

[14] M. Pardas and E. Sayrol, "Motion estimation based tracking of active contours," *Pattern Recognition Letters*, vol. 22, no. 13, pp. 1447–1456, November 2001.

[15] C.L. Lam and S.Y. Yuen, "An unbiased active contour algorithm for object tracking," *Pattern Recognition Letters*, vol. 19, no. 5-6, pp. 491–498, April 1998.

[16] J. Denzler and H. Niemann, "Active rays: Polar-transformed active contours for real-time contour tracking," *Real-Time Imaging*, vol. 5, no. 3, pp. 203–213, June 1999.

[17] Y. Chen, T.S. Huang, and Y. Rui, "Optimal radial contour tracking by dynamic programming," in *IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[18] N. Vandenbroucke, L. Macaire, C. Vieren, and J.G. Postaire, "Contribution of a color classifier to soccer players tracking with snakes," in *IEEE International Conference on System, Man and Cybernetics*, Orlando, FL, 1997, vol. 4, pp. 3660–3665.

[19] S. Lefèvre, C. Fluck, B. Maillard, and N. Vincent, "A fast snake-based method to track football players," in *IAPR International Workshop on Machine Vision Applications*, Tokyo, Japan, November 2000, pp. 501–504.

[20] L.D. Cohen, "On active contour models and balloons," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 211–218, March 1991.

[21] S. Lefèvre, L. Mercier, V. Tiberghien, and N. Vincent, "Multiresolution color image segmentation applied to background extraction in outdoor images," in *IS&T European Conference on Color in Graphics, Image and Vision*, Poitiers, France, April 2002.

[22] W.P. Choi, K.M. Lam, and W.C. Siu, "An adaptative active contour model for highly irregular boundaries," *Pattern Recognition*, vol. 34, no. 2, pp. 323–331, February 2001.

[23] A.E. Grace and D. Pycock, "Multiresolution active contour models in textured stereo images," in *British Machine Vision Conference*, Edinburgh, UK, 1996, pp. 575–584.

[24] B. Leroy, I.L. Herlin, and L.D. Cohen, "Multiresolution algorithms for active contour models," in *International Conference on Analysis and Optimization of Systems*, Rocquencourt, France, 1996, pp. 58–65.

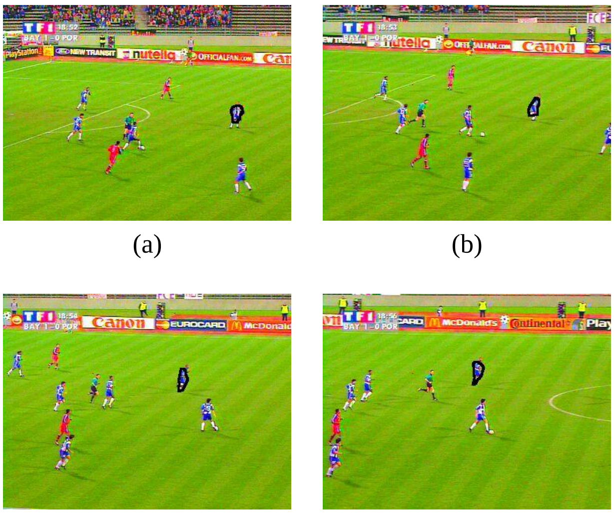
[25] N. Ray, B. Chanda, and J. Das, "A fast and flexible multiresolution snake with a definite termination criterion," *Pattern Recognition*, vol. 34, no. 7, pp. 1483–1490, July 2001.

[26] T. Gevers, S. Ghebreab, and A.W.M. Smeulders, "Color invariant snakes," in *British Machine Vision Conference*, Southampton, UK, 1998, pp. 578–588.

[27] K.P. Ngoi and J.C. Jia, "A new colour image energy for active contours in natural scenes," *Pattern Recognition Letters*, vol. 17, no. 12, pp. 1271–1277, October 1996.

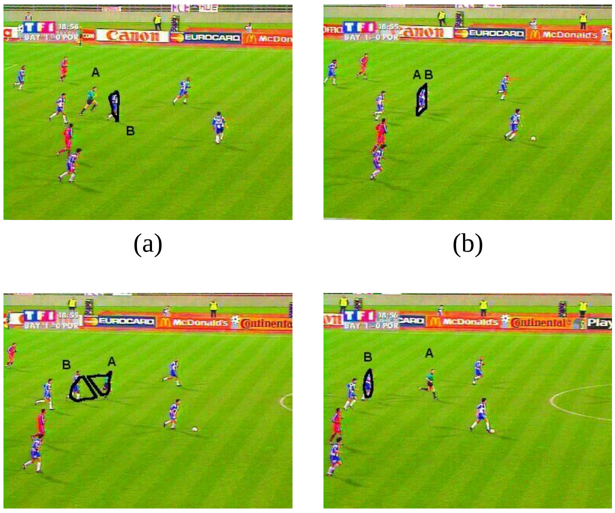
Parameter	Value
Size m (in points) of the snake (at original resolution)	16
Maximum number of iterations in the deformation (at original resolution)	30
Size (in points) of the neighbourhood scanned for local energy computation	5×5
Coefficient α for internal energy	1
Coefficient β for external energy	1
Coefficient a for continuity energy	1
Coefficient b for balloon energy	1
Coefficient c for curvature energy	1
Coefficient d for gradient energy	1
Coefficient e for colour energy	1

Table 1: Parameters used in the proposed tracking algorithm.



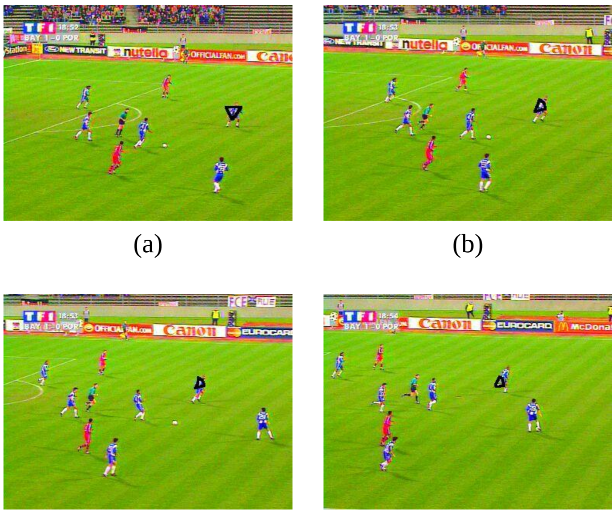
(a) (b) (c) (d)

Figure 4: Objet tracking results from a 100 frames football video sequence. Interval between two images is about 20 frames.



(a) (b) (c) (d)

Figure 5: Relevance of the scission step in case of close positions of moving objects. Two moving objects have crossing trajectories (occlusion in top right image). Scission step allows to correctly fit objects after occlusion (bottom left image).



(a) (b) (c) (d)

Figure 6: Object tracking on a multiresolution basis. Frames were analysed at a very low resolution r_4 (image size has been decreased by a factor equal to 256).