Real Time Multiple Object Tracking Based on Active Contours

Sébastien Lefèvre[1] and Nicole Vincent[2]

[1] LSIIT – University Louis Pasteur (Strasbourg I)
Parc d'Innovation, boulevard Brant, BP 10413, 67412 Illkirch Cedex, France
lefevre@lsiit.u-strasbg.fr
[2] CRIP5 – University René Descartes (Paris V)
45, rue des Saints Pères, 75270 Paris Cedex 06, France
nicole.vincent@math-info.univ-paris5.fr

Abstract. In this paper our purpose is to present some solutions to multiple object tracking in an image sequence with a real time constraint and a possible mobile camera. We propose to use active contours (or snakes) modelling. Classical active contours fail to track several objects at once, so occlusion problems are difficult to solve. The model proposed here enables some topology change for the objects concerned. Indeed a merging and a splitting phases are respectively performed when two objects become close together or move apart. Moreover, these topology changes help the tracking method to increase its robustness to noise characterized by high gradient values. In the process we have elaborated, no preprocessing nor motion estimation (which are both time consuming tasks) is required. The tracking is performed in two steps that are active contour initialisation and deformation. The process supports non-rigid objects in colour video sequences from a mobile camera. In order to take advantage of compressed formats and to speed up the process when possible, a multiresolution framework is proposed, working in the lowest-resolution frame, with respect to a quality criterion to ensure a satisfying quality of the results. The proposed method has been validated in the context of real time tracking of players in soccer game TV broadcasts. Player positions obtained can then be used in a real time analysis tool of soccer video sequences.

1 Introduction

Object tracking is a key step in automatic understanding of video sequence content. When objects are non-rigid, an appropriate tracking method should be used. Among methods that have been proposed, we can mention deformable models and active contours (or *snakes*). As we are focusing on approaches characterized by a low computational cost, we will choose active contours. Different active contour models have been proposed since the original model by Kass *et al.* [1] called snakes. This model has shown several limitations such as initialisation, optimal parameter setting, computational cost, and unability to change its topology. Some authors have proposed other models, among them we can mention geodesic contours [2] which allow to deal with topology changes. As we are focusing on real time tracking, we do not consider approaches as those based on *level sets* [3], more powerful but also with a higher computational cost. However, snakes execution in a real time framework is still a challenge.

The method we are proposing here is based on snakes but original limitations are turned away. To do so, the model considers some original energies and a two step tracking for every frame. Moreover, and contrary to other approaches, the method does not require any preprocessing nor motion estimation or compensation. Different optimisations help to obtain a real time processing.

After an introduction to the snake model, we will precise the energies and describe the two step tracking algorithm. Then we will present merging and splitting steps which let the snake change its topology and make possible a multiple object tracking. The multiresolution framework will also be described. Finally, we will illustrate our contribution with some results obtained on soccer videos.

2 Active Contours

Here we will recall main active contour models and we will give a short review of object tracking methods based on active contours. An active contour can be represented as a curve, closed or not, evolving through time. The curve is deformed in order to minimize an energy function:

$$E(v) = \int_0^1 \left[\alpha_{\text{int}} E_{\text{int}}(v(s)) + \alpha_{\text{ext}} E_{\text{ext}}(v(s))\right] ds \tag{1}$$

where E_{int} and E_{ext} represent the internal and external energies defined themselves as combinations of energies. A description of the most common energies is given in [4]. Several active contour implementations have been proposed: variational calculation, dynamic programming, or the greedy algorithm. It has been shown [5] that greedy algorithm [6] is 10 to 80 times faster than other methods, so we will focus on this approach. In the discrete domain, the definition of the snake energy function is:

$$E = \sum_{i=1}^{m} E(V^{\gamma}(i)) \qquad \text{and} \qquad V = \underset{V(i), i \in [1,m]}{\arg\min} \; E \tag{2}$$

V denotes the discrete active contour and $V(i)$ its i^{th} point from a set of m points. Contour V iteratively evolves and V^{γ} represents the active contour V at iteration γ. A minimal energy point is selected in the current point neighbourhood and a move is performed. This iterative deformation process is performed on each point of the curve, until convergence occurs.

In this section we have presented the main principles of active contours. We will now describe the tracking algorithm we are proposing, and the energies it relies on.

3 Tracking Method and Real Time Constraint

The goal of the presented method is to track in real time non-rigid objects in colour frames acquired with a moving camera. In order to minimise the computation time, we had to take several decisions which differentiate our contribution from existing methods. First, we have decided to perform active contour deformation without any preprocessing. Second, no camera motion compensation is performed. Moreover, we do not estimate the motion of the different tracked objects. Finally, gradient computation is limited to an area around the initial position of the tracked object. We will now briefly present the energies used in our model and the two steps tracking method.

3.1 Energies Definition

We consider continuity, balloon, and curvature energies that are very common internal energies. The external energy allows to link the active contour to image content, and is also composed of several energies. Here we have considered two energies based on gradient and colour information. The first tends to fit the contour to real edges of objects. We estimate the gradient of a colour image by the sum of gradients computed on the different colour components using Sobel operator. The second external force allows the snake to stay on the borders of the tracked object. To do so, it is defined using *a priori* information on the background colour features. In the case of homogeneous background, it is possible to compute the background average colour. The energy E_{col} is defined here in every point of the image as the difference between the considered point and the background average colours. In order to limit the sensitivity to noise, the value obtained is thresholded. If the background is not homogeneous, in the case of a static camera, it is possible to use a reference frame to achieve pixel colour comparison.

3.2 Two Step Tracking Method

The tracking method is composed of two steps performed successivelly on every frame: first the snake is initialised using result from previous frame, then it is deformed. The first step consists in the snake initialisation on the current frame. We enlarge a rectangle $R[0]$ with borders parallel to image borders and surronding the final snake obtained at previous frame, so *a priori* including the contour to be obtained on the current frame. We set its points regularly on the rectangle contour. Then a single object can be tracked using the forces described previously. Its position $O(t)$ at time t is computed using its previous position $O(t-1)$. Here the balloon force is used to help the snake to retract itself instead of expand. We can notice no motion estimation of the tracked object is then required. On the first frame of a video sequence, the initial rectangle is obtained from a background / foreground segmentation process [7]. In order to track in real time several objects, we have introduced several improvements using snake splitting or merging.

4 Multiple Object Tracking

The tracking method described here works even in the case of a moving camera. However, the tracking may fail if several moving objects have close spatial positions, more precisely when tracking O_i object the process fails if:

$$\exists j \; j \neq i \quad / \quad O_j(t) \subset R[O_i(t-1)] \tag{3}$$

The initial snake (rectangle) will then include both objects O_i and O_j (figure 1). After a brief review of existing approaches considering multiple objects, we will describe more precisely our solution. Therefore, in order to deal with multiple objects in the video sequence, it is necessary to bring to the model the ability to change its topology.

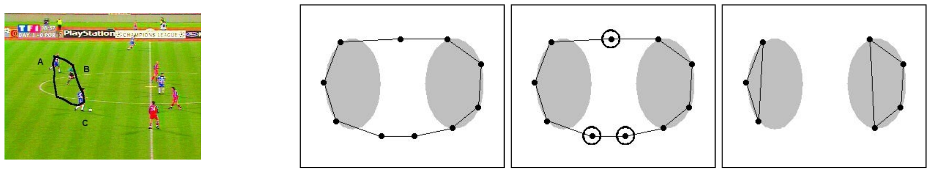

Fig. 1. Tracking failure with constant topology (left). Main steps in the splitting process (right).

4.1 Some Approaches Allowing Topology Changes

Cheung *et al.* [8] distinguish between methods with explicit use of a *split-and-merge* technique and those based on a topology-free representation as *level set*. In the T-snakes [9] of McInerney and Terzopoulos, a binary grid is associated with the image. From the grid points they determine the local positions of the topology changes to be performed. Velasco and Marroquin [10] initialise the snakes from pixels characterized by highest gradient values. Snakes are merged if their points have close positions. Ji and Yan [11] study the loops present in the contour. The procedure introduced by Choi *et al.* in [12] compares energies with a threshold at every iteration. Perera *et al.* [13] check the topological validity of the snake at every iteration. Delingette and Montagnat proposed in [14] to study crossings between two contours, and then to apply some topological operations to merge two snakes or to create new ones.

4.2 Justifications

Let us formalise the problem to be solved here in order to bring it a solution. Let us note F_t the shape of interest in the t frame. It is tracked by a snake, noted V_t. When an occlusion phenomenon occurs, the shape F_t actually represents two objects. Let us consider that the occlusion is finished at time $t+1$. We are then in presence of several shapes. We limit ourself to the case of two disjoint shapes F^1_{t+1} and F^2_{t+1}. Nevertheless, the same arguments can hold when more than two shapes are present.

The properties of these two shapes F^1_{t+1} and F^2_{t+1} can be stated as:

$$F^1_{t+1} \cap F^2_{t+1} = \varnothing \tag{4}$$

$$F^1_{t+1} \cup F^2_{t+1} \subseteq F_{t+1} \tag{5}$$

However, the snake V_{t+1}, without this information, is still considering one shape F_{t+1} (see figure 1). As we are now in presence of two shapes F^1_{t+1} and F^2_{t+1}, we have to define two appropriate snakes V^1_{t+1} and V^2_{t+1}. The problem to be solved can then be expressed as the search for a transform T which splits a snake V_{t+1} into two snakes V^1_{t+1} and V^2_{t+1} modeling respectively F^1_{t+1} and F^2_{t+1}. In the same way, a merging can be performed to gather several snakes in a unique one if necessary. However, equation (5) is not an equality, so some parts of V_{t+1} can be associated neither with F^1_{t+1} nor with F^2_{t+1}. Indeed, the shape F_{t+1} may contain background in between the two disjoint shapes F^1_{t+1} and F^2_{t+1}. At the end of the splitting process, some contours V^i_{t+1} may model shapes of no interest. So it is necessary not to take them into account. To do so, we have to identify some features of the contours V^i_{t+1} to be able to take a decision.

4.3 Principle of Topology Change

From the previous formalism, several additionnal steps are necessary in the tracking algorithm: a splitting step, a decision step which will allow to keep only interesting contours, and a merging step. In order to limit the computation time, these different steps are performed only once per frame, when convergence has been achieved using previous active contour algorithm. The main steps in the splitting process are illustrated in figure 1.

The splitting goal is to divide the snake in several contours. From equation (2), the energy obtained is a minimum. As we are using a discrete and local approach, at each of the points $V(i)$ is a local optimum. As we will see further, we would like to give the same importance to internal and external energies. External energies does not always get a minimum. Indeed, they have been thresholded in order to increase robustness to noise. Then they can be uniformly equal to zero. After the snake has converged, some points can be trapped by these areas. So we propose to delete these points and to split the active contours at the positions of these incorrect points. From the left point list, each sequence of successive points is used to create a new closed contour.

The splitting step leads to define from an initial snake several new contours. But this set of new contours may contain some snakes which fit on pixels corresponding to noise or background. So it is followed by a decision step whose goal is to determine the contours of interest. Size and shape of new potential contours are involved in the criterion we define to test the pertinence of a contour noted $(\mathcal{Q})$: the area delimited by the contour V noted $\mathrm{area}(V)$ is neither too small nor too large (*i.e.* $\mathrm{area}(V) \in [s_{\mathrm{min}}, s_{\mathrm{max}}]$), and both width and height of the circumscribed rectangle are not too small.

The splitting process described previously requires the definition of a corresponding merging process. This merging step will be performed if two objects (each of them being tracked by a snake) become closer until an occlusion phenomenon is obtained. In this case, an unique snake has to be used to model these objects. The merging process is then launched when two snakes are characterized by close gravity centers.

Here we have described how a splitting / decision / merging step of the active contour allows to deal with topology changes, to increase tracking robustness, and to ensure a simultaneous tracking of several objects. In the following section, we will show how a multiresolution analysis of video sequences frames can be performed to limit the computation time of the tracking algorithm based on active contour.

5 Multiresolution Analysis

In order to limit computation time, we propose to adapt our original snake model to analyse video frames through a coarse-to-fine multiresolution framework. The multiresolution analysis is not performed until original resolution but it stops when a criterion is verified. Moreover, we automatically adapt some model parameters to the resolution.

5.1 An Incomplete Multiresolution Process

Several authors proposed to model active contours through a multiresolution framework (*e.g.* [15]). Snake evolution is then performed on a *coarse-to-fine* approach. The snake is first deformed at a low resolution r_{max}, then the result obtained is used as the initial snake which will be deformed at a finer resolution (equal to $r_{\text{max}} - 1$). This process is repeated until the original resolution.

Here the multiresolution framework considers the image instead of the snake. Every frame is analysed at different resolutions, starting from the lowest resolution, *i.e.* $r = r_{\text{max}}$. If the previous method does not allow to obtain a correct final contour according to the decision criterion $\mathcal{Q}$, the image is then analysed at a finer resolution, *i.e.* $r \leftarrow r - 1$. The size of the image increases in an exponential way. The definition and the use of a stopping criterion linked to the quality of the results limit here the number of resolutions analysed. By this way, the computation time is also limited. This choice is particularly interesting when the contour obtained at a low resolution is sufficient to process correctly the tracking task. The algorithm proposed here is able to process images at different resolutions, from original resolution $r = 0$ to the lowest resolution $r_{\text{max}} = 5$. Most often, the tracking is performed correctly all along the video sequence on images reduced with a ratio $2^{3\times 2} = 64$.

5.2 Parameter Robustness Towards Resolution Changes

In order to ensure robustness of the algorithm towards resolution changes, we made some parameters depend on the image resolution. However the energy coefficients α and the neighbourhood size Δ_s do not depend on the resolution level. The size $X_{V_0} \times Y_{V_0}$ of the initial rectangle for the snake V_0 must obviously not be constant, as a resolution decrease implies a size decrease of the objects present in the image. Noting $\underset{0}{X_{V_0}} \times \underset{0}{Y_{V_0}}$ the size of the rectangle at original resolution $r = 0$, we get:

$$\underset{r}{X_{V_0}} = \frac{\underset{0}{X_{V_0}}}{2^r} \qquad \text{and} \qquad \underset{r}{Y_{V_0}} = \frac{\underset{0}{Y_{V_0}}}{2^r} \tag{6}$$

The same evolution function can be applied to the number m of points belonging to the snake, which also depends on the number of image pixels. As neighbourhood Δ_s is constant whereas the number of image pixels is variable, the deformation process will converge more or less quickly. Finally, gradient computation properties are not resolution independent. Indeed, the successive averaging of pixels results in an image smoothing, so threshold S_{grad} has to be adaptive.

6 Results and Discussion

We have introduced different improvements which help us to deal correctly with topology changes and to limit computation time using a multiresolution analysis of video frames. In this section, we will first indicate the different parameters and we will explain how they can be set efficiently. Then we will present some results obtained with these parameters on soccer video sequences.

The proposed method has been tested on outdoor scene video sequences characterized by a relatively uniform background. The size of colour images is equal to 384×284

pixels and the acquisition framerate is 15 Hz. The snake is initially composed of m points at the original resolution ($r = 0$). This parameter m has a direct influence on both result quality and computation time. When the application does require only the object position the number of points can be decreased. At the initial resolution ($r = 0$), the number of iterations is set equal to 30. However, the contour converges most of the time before. The coefficients used to weight the different energies have all been set to 1. It contributes to limit the number of operations (multiplications) and to greatly help in parameters setting. The threshold S_{grad} used in gradient computation has been set to 500 at the original resolution. It will be compared with the sum of gradient modules computed on colour channels with the Sobel operator. In this context, the computation time required on a PC 1.7 GHz is about 35 milliseconds per frame.

Figures 2 and 3 illustrate the tracking of non-rigid objects (soccer players) during a video sequence. The algorithm enables us to track a moving object in a moving environment, without object motion estimation nor camera motion compensation. Figure 2 illustrates the principle of the splitting and merging steps. So it is possible to track independently the different objects present in the scene. However, the sensitivity of the active contour model to a complex background (containing some pixels characterized by high gradient values) stays high. The multiresolution analysis described in the previous section is illustrated in figure 3. The resolution leads to an image size 256 times lower than the original one. We can observe the lack of precision in the snake shape.

7 Conclusion

In this article, we dealt with the problem of non-rigid object tracking using snakes. Our tracking method can be performed in real time on colour video sequences acquired with a moving camera. The method has been validated on TV broadcast of soccer games.

In order to limit the sensitivity of the model to initialisation settings, our original approach initialises a rectangular snake, and then reduces it around the object. So the tracking is robust to initialisation conditions. In order to deal with topology changes, we have introduced a splitting process, which allows to track different objects. Finally, the constraint which is the hardest to take into account is the computation time. We have combined different optimisation techniques: gradient is computed only once per frame and only on the area of interest, costly processing are not performed (global filtering or preprocessing, object motion estimation, camera motion compensation) and the images are analysed through a multiresolution framework.

We would like now to involve in our model some more robust colour or texture energies. We also consider to implement the proposed algorithm on a multiprocessor workstation in order to further limit the required computation time.

References

1. Kass, M., Witkins, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1988) 321–331
2. Paragios, N.: Geodesic Active Regions and Level Set methods: Contributions and Applications in Artificial Vision. Phd dissertation, Université de Nice Sophia-Antipolis (2000)

3. Sethian, J.: Level Set Methods and Fast Marching Methods. Cambridge Univ. Press (1999)
4. Davison, N., Eviatar, H., Somorjai, R.: Snakes simplified. Pattern Recognition **33** (2000) 1651–1664
5. Denzler, J., Niemann, H.: Evaluating the performance of active contours models for real-time object tracking. In: Asian Conference on Computer Vision, Singapore (1995) 341–345
6. Williams, D., Shah, M.: A fast algorithm for active contours and curvature estimation. Computer Vision, Graphics and Image Processing: Image Understanding **55** (1992) 14–26
7. Lefèvre, S., Mercier, L., Tiberghien, V., Vincent, N.: 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 (2002) 363–367
8. Cheung, K., Yeung, D., Chin, R.: On deformable models for visual pattern recognition. Pattern Recognition **35** (2002) 1507–1526
9. McInerney, T., Terzopoulos, D.: T-snakes: Topology adaptive snakes. Medical Image Analysis **4** (2000) 73–91
10. Velasco, F., Marroquin, J.: Growing snakes: Active contours for complex topologies. Pattern Recognition **36** (2003) 475–482
11. Ji, L., Yan, Y.: Loop-free snakes for highly irregular object shapes. Pattern Recognition Letters **23** (2002) 579–591
12. Choi, W., Lam, K., Siu, W.: An adaptative active contour model for highly irregular boundaries. Pattern Recognition **34** (2001) 323–331
13. Perera, A., Tsai, C., Flatland, R., Stewart, C.: Maintaining valid topology with active contours: Theory and application. In: CVPR, USA (2000) 496–502
14. Delingette, H., Montagnat, J.: Shape and topology constraints on parametric active contours. Computer Vision and Image Understanding **83** (2001) 140–171
15. Ray, N., Chanda, B., Das, J.: A fast and flexible multiresolution snake with a definite termination criterion. Pattern Recognition **34** (2001) 1483–1490

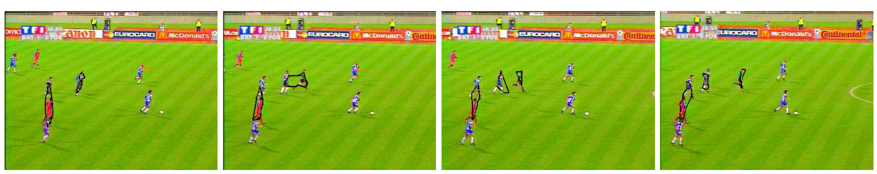

Fig. 2. Interest of splitting/merging steps in the case of close objects and temporary occlusions.

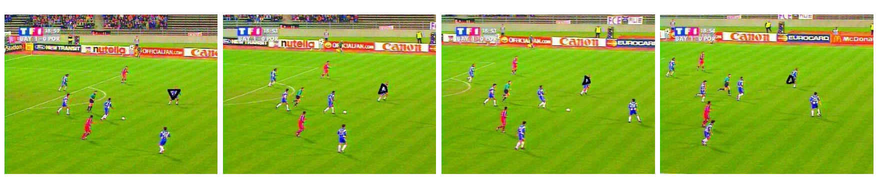

Fig. 3. Non-rigid object tracking at a resolution 256 times lower than the original one.