Tracking multiple non-rigid objects in video sequences

François Brémond and Monique Thonnat

Abstract—
This paper presents a method to track multiple non-rigid objects in video sequences. First, we present related works on tracking methods. Second, we describe our proposed approach. We use the notion of target to represent the perception of object motion. To handle the particularities of non-rigid objects we define a target as a individually tracked moving region or as a group of moving regions globally tracked. Then we explain how to compute the trajectory of a target and how to compute the correspondences between known targets and moving regions newly detected. In the case of an ambiguous correspondence we define a compound target to freeze the associations between targets and moving regions until a more accurate information is available. Finally we provide an example to illustrate the way we have implemented the proposed tracking method for video-surveillance applications.

Keywords— **Tracking, non-rigid objects, video sequence interpretation, cluttered scenes.**

I. Introduction

In this paper we present a method to track multiple non-rigid objects in video sequences. We use this method as a component of an interpretation system. The class of applications we are interested in is the automatic surveillance of indoor and outdoor partially structured scenes with a fixed monocular color camera. Given image sequences of a scene the interpretation system analyzes the behavior of real objects. In our case the real objects correspond either to humans or to vehicles. The interpretation system we have developed is composed of three modules. First, the image processing module detects moving regions in images. Then the tracking module associates these regions to track targets that correspond to real objects. Finally the scenario recognition module identifies targets as real objects and recognizes the scenarios relative to their behavior. In this paper we focus on the tracking module. The tracking problem is a central issue in scene interpretation because the loss of a tracked object leads to the impossibility of analyzing its behavior. This paper presents a tracking method based on appearance : we track the perception of scene object movements instead of tracking their real structure. Our goal is to obtain a robust algorithm able to cope with real-time constraints. First, we present related works on tracking methods. Second, we describe our proposed approach and the models of tracked moving regions. Third, we explain how the tracking process works and we show how to compute the correspondences between already tracked moving regions and newly detected ones. Finally we provide an example to illustrate the way we have implemented the proposed tracking method.

INRIA Sophia Antipolis, projet Orion 2004 route des Lucioles BP93 - 06902 Sophia-Antipolis Cedex, France. E-mail: Francois.Bremond@sophia inria.fr

II. Related works

Three main approaches have been developed to track objects depending on their type : whether they are rigid, non rigid or whether we have no information on their shape. For the two first approaches the goal of the tracking process is to compute carefully the correspondences between objects already tracked and the newly detected moving regions, whereas the goal of the last approach consists in computing coarsely the correspondences and in handling the situations where correspondences are ambiguous.

A. Tracking rigid objects

When objects are rigid, like manufactured objects, the tracking process can take advantage of accurate knowledge on their shape. Two methods can be used.

- A first method consists in detecting particular primitives, such as corners and edges, and in tracking these primitives from one image to another [5]. This method can be used only with objects owning numerous primitives easy to detect, like vehicles.
- A second method computes the correspondences between a 3D model of mobile objects and the 2D moving regions corresponding to their perception [6], [7]. For example, the center of the moving region and the direction of its motion are computed, then the correspondences between the line segments of the 3D model and the edges detected inside the region are established. This method is all the more reliable than the 3D model is accurate.

B. Tracking non-rigid objects

When objects are non-rigid, like humans, no accurate model of their shape is available. Instead, dynamic templates of the perception of object motion are used. These templates that we call models are regularly updated during the tracking process to compensate the evolution of the object perception. Three types of dynamic model can be used.

- The first type of model corresponds to the parameterized shape of the mobile objects. These models can be applied to rigid or non-rigid objects. For example in [3], the authors use polygons to represent the outline of vehicles. For similar applications in [8] the authors use cubic B-splines instead of polygons. In both cases these models have been successfully applied to track rigid objects. In [1] the authors extend the method to non-rigid objects. Their model is made of cubic B-splines but it also contains the authorized deformations that correspond to the outline of a walking pedestrian. By this way the outline of tracked objects can be distorted only in certain directions making the tracking process more reliable.

- The second type of model contains a template of the moving region corresponding to the detection of the object motion. This template is defined by the color distribution of the pixels belonging to the moving region. For example in [9], the authors use a color histogram and thanks to it they are able to track in the same time several football players and to cope with dynamic occlusions in certain situations. In [10], [11], the authors associate each pixel of the template to the temporal color distribution of the intensity function. Thus they obtain a robust algorithm and they manage to accurately track the motion of a person located in front of a CCD camera.
- The third type of model is also made of a template of the moving region. However this template is defined by the set of edges detected in the moving region. In [12], the authors use this template and define a distance between two sets of edges to allow them to compare parts of templates. Thanks to this model the authors are able to track a man even if he is partially occluded.

Therefore all these methods allow the tracking process to compute reliable correspondences between already tracked objects and newly detected moving regions. However these methods require strong conditions to work efficiently.

C. *Tracking without object model*

When no *a priori* model of objects is available, the tracking process can only use the coordinates of object locations to compute the correspondences between already tracked objects and newly detected ones. As ambiguous correspondences may arise, several methods have been proposed to solve these ambiguities.

- For example the method of *"Multiple Hypothesis Tracking"* (MHT) generates hypotheses to perform all the possible combinations of correspondences. These hypotheses define different worlds where the correspondences are coherent. When a hypothesis becomes incoherent the associate correspondences are discarded. To avoid a combinatorial explosion only a few levels of hypotheses are computed in real-world applications. In [2], the authors propose an efficient implementation of MHT.
- The *beam search method* also proposes a mechanism to handle ambiguities [4]. It duplicates all the ambiguous tracked objects and makes the correspondences with the associate newly detected objects. Then this method consists in tracking all these objects and verifies whether or not they are coherent at every new frame arrival. If their tracking is coherent, the correspondences relative to these objects are considered as the true ones. If it is not the case, the relative objects are discarded. Therefore no coherence has to be maintained during the tracking process, however we never know if a tracked object is really a true one or only if it is a duplicated object which incoherence could not be verified.

These methods are two examples of tracking methods that can be applied to any kind of objects. More generally all these methods have the same characteristics : as they do not use *a priori* knowledge on objects, they cannot compute accurate correspondences and may lead to tracking errors.

III. The proposed approach

To handle the particularities of non-rigid objects we propose an appearance-based approach for the tracking method.

A. *Moving regions and non-rigid mobile objects*

In our interpretation system the moving regions are computed by the detection module. As the camera is fixed, this module subtracts the current image from the background image which is regularly updated to compensate for illumination change. Because of the acquisition conditions the tracking module has to face two kinds of problems : it has to fix detection problems (e.g. shadows, reflections, blinking lights) and to take care of common tracking problems (e.g. occlusions, merges of tracks, appearances of new tracks). Unfortunately in our applications the conditions of image acquisition, such as image resolution, are poor making these problems a crucial issue. Moreover the model of mobile object is inaccurate because of the non-rigid nature of some objects like humans, avoiding for example the use of information characterizing their shape.

For these reasons a moving region can either correspond to the perception of a noise (e.g. a shadow), of a real object (e.g. a person), of a part of a real object (e.g. the head of a person) or of a group of real objects (e.g. a crowd). Therefore we decide to base the tracking process on the appearance of the mobile objects (i.e. the moving regions) instead of their complete structure. For example in the case of a man whose head is the only visible part, we prefer tracking the moving region corresponding to his head rather than reconstructing and tracking the complete body. Thus, a **target** of the tracking process is any individually tracked moving region or any group of moving regions which are closed to each others and which are globally tracked. This definition allows us to track any mobile object despite detection errors.

B. *Target model*

To improve the tracking of non-rigid objects a solution as suggested in [3], [1], [11] is to define a moving region as a set of primitives (e.g. corners, contour edges, color regions) and to track globally the moving region by combining all its primitives. So the loss of one primitive is compensated by the use of the other primitives. However two problems may arise from this method. First, the primitives are not always available especially if the resolution is low. For example when tracking humans and more generally non manufactured objects, it is difficult to detect corners in the moving region corresponding to the perception of their motion. Moreover in cases where images are really noisy, taking into account badly detected primitives can distort the global computation of the track. To be less primitive-dependent we propose to define and to track generic points that represent the tracked moving region (or a group of moving regions). Thus, we define the target model thanks to the height and width of the bounding box surrounding the moving regions associated to the target and thanks to five generic points as shown on figure 1. The height and width are average values regularly updated during the process. The five generic points are defined as the middle of the sides of the bounding box and as its center. They are tracked separately and the point that best matches the newly detected moving region defines the track of the global object.

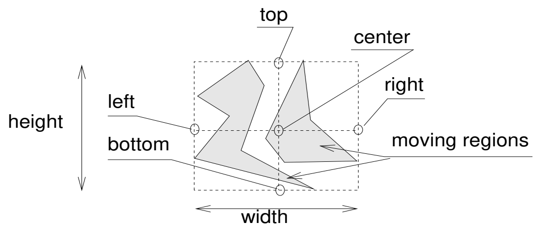

Fig. 1. The five generic points (center, top, bottom, left, right) define the model of a target corresponding to a group of moving regions.

This method has three main interests. First, the target model allows the tracking process to be less sensitive to partial occlusion. If a point has a bad match, then it is not taken into account and it gives a clue on the presence of a possible occlusion. The value of this point is then computed thanks to the other generic points and the average height and width of the target. This model is called dynamic because the shape of the bounding box associated to the target is tuned at every new frame arrival. Second, the target model is minimal and generic. It can be defined on any moving region (or on any group of moving regions) and can be used to track either rigid or non rigid objects. To improve the accuracy of the processing, the model can be extended with additional primitives (e.g. color regions) as soon as the application can afford it. Third, as generic points are not directly connected with reality they allow us to track a target without having to know what it represents. Thus, we separate the tracking module from the scenario recognition module in charge of identifying real objects.

C. *Tracking improvement*

To handle some tracking problems we also use two kinds of additional information. First, we use contextual information on the static environment to adapt the tracking process to the current scene [13]. We have divided the spatial structure of the environment into a tessellation of polygonal zones, that provides us with information on the tracked moving regions. For example, an *"exit zone"* indicates that targets inside this zone are liable to leave the scene leading to a potential termination of their tracks. Thus, we call contextual information of a target the information associated to the zone where the target is located. This information allows us to manage the tracks of targets, like its initiation and termination, and to solve tracking problems due to the environment, like static occlusions and zones with a patterned background leading to a lack of intensity contrast.

Second, we use information computed by the scenario recognition module to improve the tracking process. This module holds a specific knowledge related to the application and it is in charge of identifying mobile objects. The scenario recognition module can then indicate to the tracking process the uninteresting and interesting targets. For example, a target corresponding to a reflection is uninteresting and it has to be discarded, whereas a target corresponding to a mobile object that behaves abnormally is interesting for the application and its track has not to be lost. So this module guides the tracking process to be more efficient.

Therefore, thanks to the context and the scenario recognition module, we can adapt the tracking process to the scene environment and to the application.

IV. Tracking process

The tracking of moving regions is performed in a prediction-matching-update loop. At time t, the tracking module predicts the next location of targets tracked at time *t-1* and tries to match these predictions with the moving regions detected at time t. Once the correspondences of targets are established, the tracking module updates their tracks.

A. *Prediction of the target location*

In the literature probabilistic methods (e.g. Kalman filters or conditional probabilities) are generally used to compute the motion of rigid objects and to predict their new locations [14], [3], [4]. However we think that these methods do not perfectly suit to the tracking of humans because they require accurate models of motion and noise and because they do not allow radical motion changes. For example in [10], the author needs to combine 15 filters to model all the movements of a man while driving a car, like turning the wheel. In our approach we compute the predicted location of a target thanks to its trajectory. To allow changes we suppose that the motion of a target is piecewise linear and we represent its trajectory by a polygonal approximation. We compute short segments of line for the trajectory to consider that the speed of the target is constant on segments. So the trajectory gathers an approximation of all past locations of the target. The new location of a target is computed during the matching process between the target and a moving region detected in the current frame. The location at time t of every generic point of the target is estimated by extending the target trajectory with the generic point location at time *t-1*. To extend the trajectory we compute the motion vector using just the last segment if the trajectory is reliable or using more segments if it is not the case. Then, every estimated location is matched with the location of the corresponding generic point of the moving region. The target generic point with the best match is chosen to compute the location of the target center at time t thanks to the target height and width average values. Thus trajectories allow us to predict the location of targets.

B. Ambiguity matrix

The matching process compares the predicted location of targets with the location of newly detected moving regions through the use of an ambiguity matrix, also called the association matrix. In this matrix the columns represent the targets and the rows represent the moving regions. The matrix elements measure the distance between targets and moving regions which is defined as followed :

if intersection(moving region, target) $\neq \emptyset$
then distance $= 1 - \frac{surface\ intersection}{surface\ union}$(moving region, target)
else distance $= 1 + \frac{Euclidean\ distance}{maximal\ size}$(moving region, target)

This definition does not correspond to a regular distance (the triangular inequality is not satisfied), but it allows us to balance the distance between moving regions and targets through their size. So the ambiguity matrix gives the number of moving regions that are closed to the predicted location of each target. A moving region is said to be closed to an other region if their distance is below a certain threshold and the difference between the size of the two regions does not exceed another threshold. We choose large thresholds so that the tracking algorithm do not depend on the image sequence. According to this number we define four states for a target :

- **Visible** : if the target corresponds to one moving region and the moving region has no other correspondence.
- **Lost** : if the target does not correspond to any moving region.
- **Occluded** : if the target does not correspond to any moving region and if the contextual information of the target indicates the possibility of a static occlusion.
- **Ambiguous** : if the target corresponds to several moving regions or if the target corresponds to one moving region having other correspondences.

Thanks to the ambiguity matrix the correspondences are globally computed. The remaining task of the matching process is to solve the correspondences of the ambiguous targets.

C. Solving ambiguous correspondence

The way the tracking algorithm solves the ambiguous correspondences between targets and moving regions determines the robustness of tracking algorithms. To solve the ambiguous correspondences we define compound targets. We call a **compound target** a target that models an ambiguity. It allows us to globally track a set of temporary targets while maintaining and freezing a set of ambiguous targets. **Temporary targets** represent the newly detected moving regions which are associated to the ambiguous correspondence, whereas **ambiguous targets** represent the history of the tracking before the ambiguity has been met. The goal of the tracking algorithm is to associate the ambiguous targets with the temporary ones as soon as more accurate information is available. Meanwhile the compound target is used to freeze the associations. According to the type of ambiguity we define three types of target, illustrated on figure 2 :

- **Split** : if one ambiguous target corresponds to several newly detected moving regions, temporary targets are initialized for each moving region and a target of type split is created. The split target gathers the temporary targets and the ambiguous target.
- **Merge** : if several ambiguous targets correspond to one newly detected moving region, a temporary target is initialized with the moving region and a target of type merge is created. The merge target gathers the temporary target and the ambiguous targets.
- **Mixed** : if several ambiguous targets correspond to several newly detected moving regions, temporary targets are initialized for each moving region and a target of type mixed is created. The mixed target gathers the temporary targets and the ambiguous targets.

The tracking algorithm is able to solve the ambiguity when a temporary target of the compound target moves away from the other temporary targets. In this case the tracking algorithm tries to find the ambiguous target that corresponds to the temporary target moving away. There are two mechanisms to find the ambiguous target according to the arrangement of ambiguous targets in the compound target. The first mechanism is used when ambiguous targets verify the *rigidity* condition. If ambiguous targets have the same parallel motion, then these targets are supposed to keep their spatial ordering inside the compound target. So the temporary target moving away is associated with the ambiguous target with the same ordering number. The second mechanism is used when ambiguous targets have distinct motions. In this case a local matrix of ambiguity is used to compute the correspondences between ambiguous and temporary targets. If these arrangements are not verified, the moving away temporary target is not associated to any ambiguous target. With these mechanisms an ambiguous target can be retrieved and the corresponding ambiguous correspondence can be solved without approximation.

When the compound target modeling an ambiguity is facing a second ambiguity, the tracking algorithm has to solve the first ambiguity to avoid an combinatorial explosion. So a second compound target has to be built to model the second ambiguity and the ambiguous targets of the first compound target have to be transformed into ambiguous targets of the second compound target. This transformation consists in retrieving the missing locations of the ambiguous targets during the time interval between the occurrence of the two ambiguities. Then we propose to approximate the missing locations by using the locations of the first compound target. The accuracy of the approximation

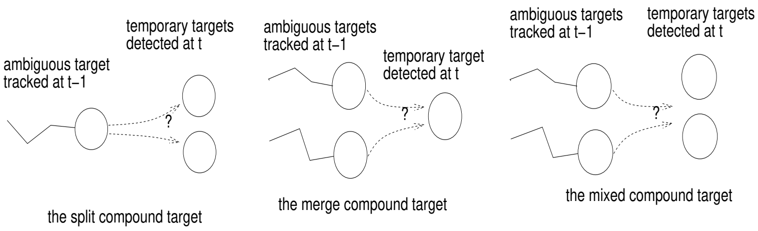

Fig. 2. The three types of compound target.

depends on the type of the first compound target. With the split type there is no approximation because the split target contains only one ambiguous target. The merge type gives a better approximation than the mixed type, because with a merge type ambiguous targets are usually closer to each others. Therefore all ambiguities are solved with more and less approximation.

The method of the compound targets can be related to the *Multiple Hypothesis Tracking* method [2] and to the *beam search* method [4]. Unlike the MHT the compound target method does not keep all the past information on the correspondences but combines and approximates them. The compound target method differs also from the beam search method in that the compound target method does not track incoherent targets. It solves ambiguities as late as possible. Thus the compound target method needs to maintain less information on past that the two other methods but approximates the computation of correspondences. The comparative results depend on the frequency of the situations where the compound target method has to make approximations.

D. Update of non ambiguous correspondence

The update of a non ambiguous target depends on its state. If the target is visible, its trajectory is extended with the location of the associate moving region. If the target has been lost for a short time or if it is occluded, we suspend its track and we try to retrieve its correspondence in next frames. If the target has been lost for a time long enough and if the context indicates that the target is in an exit zone, then the track is considered as finished and the target is discarded. If the target has been lost for a time long enough and if it has no particular contextual information, then the target is supposed to be definitively lost and it is discarded. When all targets are updated, new targets are initialized with the moving regions that have no correspondence with a known target.

When all correspondences have been taken care, the targets are processed by the scenario recognition module to try to analyze whether or not targets correspond to real objects. Using *a priori* knowledge on real objects a belief coefficient is established for each target in the framework of fuzzy sets [15]. Then, in a feedback stage the tracking module eliminates all targets with a belief coefficient not high enough. This feedback stage provides an additional information to discard the targets that correspond to detection errors like reflections. The method of the belief coefficient can be related to the method of the support of existence [4] which differs on the nature of the additional information. The support of existence relies on tracking features (e.g. the target life time) whereas the belief coefficient relies on the behavior analysis of targets. As this coefficient is computed by the scenario recognition module, it is not described in this paper.

V. Example of tracking

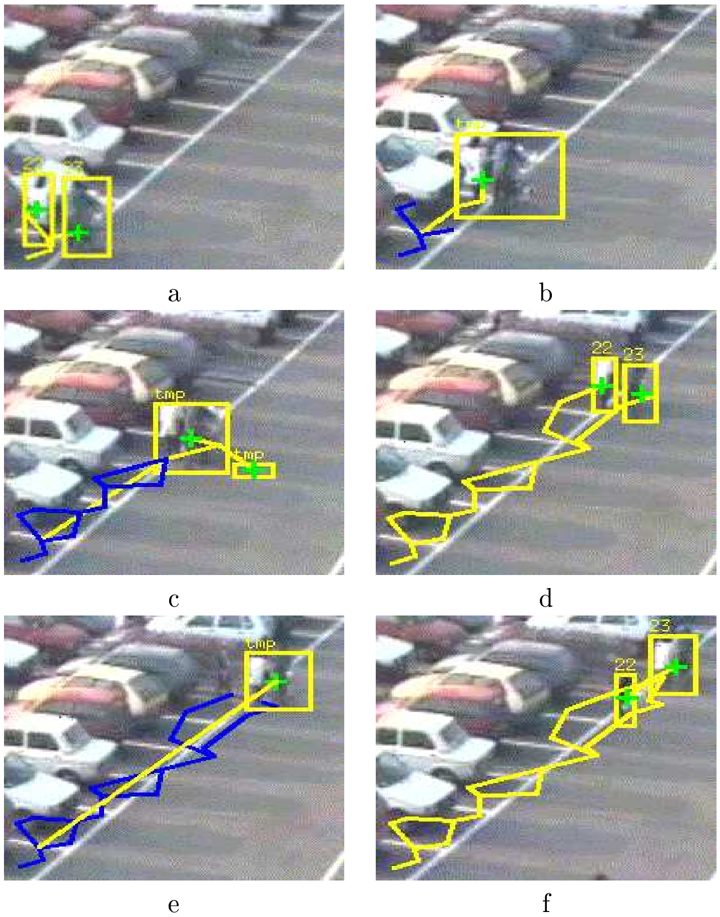

Fig. 3. This figure shows two persons being tracked. On frames a and d they are tracked as individual, whereas on the other frames they are tracked as a group. The figure shows also the trajectory of the group and of each person.

In the framework of a video-surveillance application we have developed a system implementing in C++ language the tracking algorithm described in this paper. To validate our system we use image sequences taken in the framework of the European Esprit PASSWORDS project. The scenes depicted by the sequences are related to supermarket, metro or car park areas. The sequences contain between 200 and 1200 frames. For a color image frame at 512x512 resolution the average time of the global processing is 2.5 seconds on a Sun Sparc station SS-10, whereas the average time of the tracking process alone is around 0.2 seconds. The success of the tracking algorithm depends on the quality of video sequences. With little noise the tracking algorithm is able to correctly track real objects. For example figure 3 shows two persons (numbered 22 and 23) walking together splitting (frame a) and merging (frame b) along their way. Despite the shadows that add noise (frame c) the two persons are still tracked (frame d). Frame e shows that thanks to compound targets the tracks of the group and of each person are not lost. However on frame f the target (numbered 22) is mislabeled because a third person is going out from cars (this person was not detected on frame e) and simultaneously is merging with the two first persons.

VI. Conclusion

This paper presents a tracking method that gathers several characteristics:

- The method is based on the tracking of the appearance of scene objects instead of their real structure. This approach allows us to handle several cases of partial static occlusion. In particular, it helps in tracking scene objects even if they are partially detected.
- The method uses elementary dynamic models of targets that need no *a priori* knowledge on scene objects. Because these models are elementary we can take into account any kind of target, whatever are its correspondences with scene objects.
- The method allows us to solve several cases of ambiguous correspondences. Thanks to compound targets we are able to freeze the association and the tracking of ambiguous targets until more accurate information is available.
- The method systematically uses two other types of information: contextual information and information computed by the scenario recognition module. For example this information allows us to handle total occlusion and to discard targets that correspond to noise.

In the scientific community researchers usually emphasize one of these characteristics when developing tracking methods. For example in [12], the authors use accurate dynamic models of targets to compute their correspondence, but they do not handle ambiguous correspondences. Our approach consists in combining all these characteristics in order to obtain a more reliable algorithm to develop real applications: to handle real conditions of image acquisition and to handle various types of video sequences. Thus this proposed tracking method allows us to identify scene objects and to analyze their behavior on long temporal video sequences.

We consider that the results of our tracking method are satisfactory besides that in some noisy video sequences targets are still lost. There are many ways to improve the different stages of the tracking algorithm depending on the type of video sequences. Our future works will consist in adapting the global tracking processing to these various types. For example we plan to take advantage of specific application conditions to enhance the processing.

References

[1] A. Baumberg and D. Hogg, "An adaptive eigenshape model," in *proc. of the British Machine Vision Conference (BMVC)*, Birmingham, Sept. 1995.

[2] I. Cox and S. Hingorani, "An efficient implementation of reid's Multiple Hypothesis Tracking algorithm and its evaluation for the purpose of visual tracking," in *IEEE Transactions on pattern analysis and machine intelligence*, Sept. 1996, vol. 18.

[3] F. Meyer and P. Bouthemy, "Region-based tracking in an image sequence," in *Proc. of European Conference on Computer Vision (ECCV)*, May 1992, pp. 476–484.

[4] Z. Zhang, "Token tracking in a cluttered scene," Research report 2072, I.N.R.I.A., Sophia Antipolis, Oct. 1993.

[5] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation," *Image and Vision Computing*, vol. 13, pp. 695–703, Nov. 1995.

[6] L. Du, G. Sullivan, and K. Baker, "Quantitative analysis of the viewpoint consistency constraint in model-based vision," in *Proc. of the International Conference on Computer Vision 93, Berlin, Germany*, May 1993, pp. 632–639.

[7] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *International Journal of Computer Vision*, vol. 10, no. 3, pp. 257–281, 1993.

[8] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer, "Tracking complex primitives in an image sequence," in *proc. of the ICCV'94, Jerusalem*, 1994.

[9] S. Choi, Y. Seo, H. Kim, and K. Hong, "Where are the ball and players? Soccer game analysis with color-based tracking and image mosaik," in *ICIAP'97*, 1997, to appear.

[10] A. Pentland, "Machine understanding of human action," in *proc. of the 7th Int'l Forum on Frontier of Telecommunication Technology*, Tokyo, Nov. 1995.

[11] A. Azarbayejani, C. Waren, and A. Pentland, "Real-time 3D tracking of the human body," in *Proc. of IMAGE'COM 96*, Bordeaux, May 1996.

[12] P. Huttenlocker and W. Rucklidge, "Tracking non-rigid objects in complex scenes," in *proc. of Int'l Conf. on Computer Vision (ICCV)*, Berlin, Sept. 1992.

[13] F. Brémond and M. Thonnat, "A context representation for surveillance systems," in *Proc. of the Workshop on Conceptual Descriptions from Images at the European Conference on Computer Vision (ECCV)*, Cambridge, April 1996.

[14] Y. Bar-Shalom and T. Fortmann, *Tracking and data association*, Academic press, London, 1988.

[15] F. Brémond and M. Thonnat, "Interprétation de séquences d'images et incertitude," in *Proc. of the Rencontres sur la Logique Floue et ses Applications (LFA)*, Nancy, Dec. 1996.