Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 274349, 21 pages
doi:10.1155/2008/274349

Research Article

Feature Classification for Robust Shape-Based Collaborative Tracking and Model Updating

M. Asadi, F. Monti, and C. S. Regazzoni

Department of Biophysical and Electronic Engineering, University of Genoa, Via All'Opera Pia 11a, 16145 Genoa, Italy

Correspondence should be addressed to M. Asadi, asadi@dibe.unige.it

Received 14 November 2007; Revised 27 March 2008; Accepted 10 July 2008

Recommended by Fatih Porikli

A new collaborative tracking approach is introduced which takes advantage of classified features. The core of this tracker is a single tracker that is able to detect occlusions and classify features contributing in localizing the object. Features are classified in four classes: good, suspicious, malicious, and neutral. Good features are estimated to be parts of the object with a high degree of confidence. Suspicious ones have a lower, yet significantly high, degree of confidence to be a part of the object. Malicious features are estimated to be generated by clutter, while neutral features are characterized with not a sufficient level of uncertainty to be assigned to the tracked object. When there is no occlusion, the single tracker acts alone, and the feature classification module helps it to overcome distracters such as still objects or little clutter in the scene. When more than one desired moving objects bounding boxes are close enough, the collaborative tracker is activated and it exploits the advantages of the classified features to localize each object precisely as well as updating the objects shape models more precisely by assigning again the classified features to the objects. The experimental results show successful tracking compared with the collaborative tracker that does not use the classified features. Moreover, more precise updated object shape models will be shown.

1. INTRODUCTION

Target tracking in complex scenes is an open problem in many emerging applications, such as visual surveillance, robotics, enhanced video conferencing, and sport video highlighting. It is one of the key issues in the video analysis chain. This is because the motion information of all objects in the scene can be fed into higher-level modules of the system that are in charge of behavior understanding. To this end, the tracking algorithm must be able to maintain the identities of the objects.

Maintaining the track of an object during an interaction is a difficult task mainly due to the difficulty in segmenting object appearance features. This problem affects both locations and models of objects. The vast majority of tracking algorithms solve this problem by disabling the model updating procedure in case of an interaction. However, the drawback of these methods arises in case of a change in objects appearance during occlusion.

While in case of little clutter and few partial occlusions it is possible to classify features [1, 2], in case of heavy interaction between objects, sharing trackers information can help to avoid the coalescence problem [3].

In this work, a method is proposed to solve these problems by integrating an algorithm for feature classification, which helps in clutter rejection, in an algorithm for the simultaneous and collaborative tracking of multiple objects. To this end, the Bayesian framework developed in [2] for shape and motion tracking is used as the core of the single object tracker. This framework was shown to be a suboptimal solution with respect to the single-target-tracking problem, where a posterior probabilities of the object position and the object shape model are maximized separately and suboptimally [2]. When an interaction occurs among some objects, a newly developed collaborative algorithm, capable of feature classification, is activated. The classified features are revised using a collaborative approach based on the rationale that each feature belongs to only one object [4].

The contribution of this paper is to introduce a collaborative tracking approach which is capable of feature classification. This contribution can be seen as three major points.

(1) Revising and refining the classified features. A collaborative framework is developed that is able to revise and refine classes of features that have been classified by the single object tracker.

(2) Collaborative position estimation. The performance of the collaborative tracker is improved using the refined classes.

(3) Collaborative shape updating. While the methods available in literature are mainly interested in the collaborative estimation, the proposed method implements a collaborative appearance updating.

The rest of the paper is organized as follows. Section 2 discusses the related works. Section 3 describes the single-tracking algorithm and its Bayesian origin. In Section 4, the collaborative approach is described. Experimental results are presented in Section 5. Finally in Section 6 some concluding remarks are provided.

2. RELATED WORK

Simultaneous tracking of visual objects is a challenging problem that has been approached in a number of different ways. A common approach to solve the problem is the Merge-Split approach: in an interaction, the overlapping objects are considered as a single entity. When they separate again, the trackers are reassigned to each object [5, 6]. The main drawbacks of this approach are the loss of identities and the impossibility of updating the object model.

To avoid this problem, the objects should be tracked and segmented also during occlusion. In [7], multiple objects are tracked using multiple independent particle filters. In case of independent trackers, if two or more objects come into proximity, two common problems may occur: "labeling problem" (the identities of two objects are inverted) and "coalescence problem" (one object hijacks more than one tracker). Moreover, the observations of objects that come into proximity are usually confused and it is difficult to learn the object model correctly. In [8] humans are tracked using a priori target model and a fixed 3D model of the scene. This allows the assignment of the observations using depth ordering. Another common approach is to use a joint-state space representation that describes contemporarily the joint state of all objects in the scene [9–13]. Okuma et al. [11] use a single particle filter tracking framework along with a mixture density model as well as an offline learned Adaboost detector. Isard and MacCormick [12] model persons as cylinders to model the 3D interactions. Although the above-mentioned approaches can describe the occlusion among targets correctly, they have to model all states with exponential complexity without considering that some trackers may be independent. In the last few years, new approaches have been proposed to solve the problem of the exponential complexity [5, 13, 14]. Li et al. [13] solve the complexity problem using a cascade particle filter. While good results are reported also in low-frame rate video, their method needs an offline learned detector and hence it is not useful when there is no a priori information about the objects class. In [5], independent trackers are made collaborative and distributed using a particle filter framework. Moreover, an inertial potential model is used to predict the tracker motion. It solves the "coalescence problem," but since global features are used without any depth ordering, updating is not feasible during occlusion. In [14], a belief propagation framework is used to collaboratively track multiple interacting objects. Again, the target model is learned offline.

In [15], the authors use an appearance-based reasoning to track two faces (modeled as multiple view templates) during occlusion by estimating the occlusion relation (depth ordering). This framework seems limited to two objects and since it needs multiple view templates and the model is not updated during tracking, it is not useful when there is no a priori information about the targets. In [16], three Markov random fields (MRFs) are coupled to solve the tracking problem: a field for the joint state of multiple targets; a binary random process for the existence of each individual target; and a binary random process for the occlusion of each dual adjacent target. The inference in the MRF is solved by using particle filtering. This approach is also limited to a predefined class of objects.

3. SINGLE-OBJECT TRACKER AND BAYESIAN FRAMEWORK

The role of the single tracker—introduced in [2]—is to estimate the current state of an object, given its previous state and current observations. To this end, a Bayesian framework is presented in [2]. The framework also is briefly introduced here.

Initialization

A desired object is specified with a bounding box. Then, all corners, say M, inside the bounding box are extracted and they are considered as the object features. They are shown as $\mathbf{X}_{c,t} = \{\mathbf{X}_{c,t}^m\}_{1\le m\le M} = \{(x_t^m, y_t^m)\}_{1\le m\le M}$ where the pair (x_t^m, y_t^m) is the absolute coordinates of corner m and the subscript c denotes *corner*. A reference point, for example, the center of the bounding box, is chosen as the object position. In addition, an initial *persistency* value P_I is assigned to each corner. It is used to show the consistency of that corner during time.

Target model

The object shape model is composed of two elements: $\mathbf{X}_{s,t} = \{\mathbf{X}_{s,t}^m\}_{1\le m\le M} = \{[\mathbf{DX}_{c,t}^m, P_t^m]\}_{1\le m\le M}$. The element $\mathbf{DX}_{c,t}^m = \mathbf{X}_{c,t}^m - \mathbf{X}_{p,t}$ is the relative coordinates of corner m with respect to the object position $\mathbf{X}_{p,t} = (x_t^{\text{ref}}, y_t^{\text{ref}})$. Therefore, the object status at time t is defined as $\mathbf{X}_t = \{\mathbf{X}_{s,t}, \mathbf{X}_{p,t}\}$.

Observations

The observations set $\mathbf{Z}_t = \{\mathbf{Z}_t^n\}_{1\le n\le N} = \{[x_t^n, y_t^n]\}_{1\le n\le N}$ at any time t is composed of the coordinates in the image plane of all extracted corners inside a bounding box $\mathbf{Q}$ of the same size as the one in the last frame, centered at the last reference point $\mathbf{X}_{p,t-1}$.

Probabilistic Bayesian framework

In the probabilistic framework, the goal of the tracker is to estimate the posterior $p(\mathbf{X}_t \mid \mathbf{Z}_t, \mathbf{X}_{t-1} = \mathbf{X}^*_{t-1})$. In this paper, random variables are vectors and they are shown using bold fonts. When the value of a random variable is fixed, an asterisk is added as a superscript of the random variable. Moreover, for simplification, the fixed random variables are replaced just by their values: $p(\mathbf{X}_t \mid \mathbf{Z}_t, \mathbf{X}_{t-1} = \mathbf{X}^*_{t-1}) = p(\mathbf{X}_t \mid \mathbf{Z}_t, \mathbf{X}^*_{t-1})$. Moreover, at time t it is supposed that the probability of the variables at time $t-1$ has been fixed. The other assumption is that since Bayesian filtering propagates densities and in the current work no density or error propagation is used, the probabilities of the random variables at time $t-1$ are redefined as Kronecker delta functions, for example, $p(\mathbf{X}_{t-1}) = \delta(\mathbf{X}_{t-1} - \mathbf{X}^*_{t-1})$. Using Bayesian filtering approach and considering the independence between shape and motion one can write [2]

$$\begin{aligned} &p(\mathbf{X}_t \mid \mathbf{Z}_t, \mathbf{X}^*_{t-1}) \\ &= p(\mathbf{X}_{p,t}, \mathbf{X}_{s,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}) \\ &= p(\mathbf{X}_{s,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}_{p,t}) \cdot p(\mathbf{X}_{p,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}). \end{aligned} \tag{1}$$

Maximizing separately each of the two terms at the right-hand side of (1) provides a suboptimal solution to the problem of estimating the posterior of $\mathbf{X}_t$. The first term is the posterior probability of the shape object model (shape updating phase). The second term is the posterior probability of the object global position (object tracking).

3.1. The global position model

The posterior probability of the object global position can be factorized into a normalization factor, the position prediction model (a priori probability of the object position), and the observation model (likelihood of the object position) using the chain rule and considering the independence between shape and model [2]:

$$\begin{aligned} &p(\mathbf{X}_{p,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}) \\ &\quad = k \cdot p(\mathbf{X}_{p,t} \mid \mathbf{X}^*_{p,t-1}) \cdot p(\mathbf{Z}_t \mid \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}_{p,t}). \end{aligned} \tag{2}$$

3.1.1. The position prediction model (the global motion model)

The prediction model is selected with the rationale that an object cannot move faster than a given speed (in pixels). Moreover, defining different prediction models gives different weights to different global object positions in the plane. In this paper, a simple global motion prediction model of a uniform windowed type is used:

$$p(\mathbf{X}_{p,t} \mid \mathbf{X}^*_{p,t-1}) = \begin{cases} \dfrac{1}{W_x \cdot W_y} & \text{if } (\mathbf{X}_{p,t} - \mathbf{X}^*_{p,t-1}) \le \dfrac{\mathbf{W}}{2}, \\ 0 & \text{elsewhere,} \end{cases} \tag{3}$$

where $\mathbf{W}$ is a rectangular area $W_x \times W_y$ initially centered on $\mathbf{X}^*_{p,t-1}$. If more a priori knowledge about the object global motion is available, it will be possible to assign different probabilities to different positions inside the window using different kernels.

3.1.2. The observation model

The position observation model is defined as follows:

$$p(\mathbf{Z}_t \mid \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}_{p,t}) = \frac{1 - e^{-V_t(\mathbf{X}_{p,t}, \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1})}}{\sum_{\mathbf{Z}_t} \left(1 - e^{-V_t(\mathbf{X}_{p,t}, \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1})}\right)}, \tag{4}$$

where $V_t(\mathbf{X}_{p,t}, \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1})$ is the number of votes to a potential object position. It is defined as follows:

$$\begin{aligned} &V_t(\mathbf{X}_{p,t}, \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}) \\ &\quad = \sum_{n=1}^{N} \sum_{m=1}^{M} K_R(d_{m,n}(\mathbf{X}^m_{c,t-1} - \mathbf{X}^*_{p,t-1}, \mathbf{Z}^n_t - \mathbf{X}_{p,t})), \end{aligned} \tag{5}$$

where $d_{m,n}(\cdot)$ is the Euclidean distance metric and it evaluates the distance between a model element m and an observation element n. If this distance falls within the radius R_R of a position kernel $K_R(\cdot)$, m and n will contribute to increase the value of $V_t(\cdot)$ based on the definition of the kernel. It is possible to have different types of kernels, based on the a priori knowledge about the rigidity of desired objects. Each kernel has a different effect on the amount of the contribution [2]. Having a look at (5), it is seen that an observation element n may match with several model elements inside the kernel to contribute to a given position $\mathbf{X}_{p,t}$. The fact that a rigidity kernel is defined to allow possible distorted copies of the model elements contribute to a given position is called *regularization*. In this work, a uniform kernel is defined:

$$\begin{aligned} &K_R(d_{m,n}(\mathbf{X}^m_{c,t-1} - \mathbf{X}^*_{p,t-1}, \mathbf{Z}^n_t - \mathbf{X}_{p,t})) \\ &\quad = \begin{cases} 1 & \text{if } d_{m,n}(\mathbf{X}^m_{c,t-1} - \mathbf{X}^*_{p,t-1}, \mathbf{Z}^n_t - \mathbf{X}_{p,t}) \le R_R, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

The proposed suboptimal algorithm fixes as a solution the value $\mathbf{X}_{p,t} = \mathbf{X}^*_{p,t}$ that maximizes the product in (2).

3.1.3. The hypotheses set

To implement the object position estimation, (5) is implemented. Therefore, it provides an estimation for each point $\mathbf{X}_{p,t}$ of the probability that the global object position is coincident with $\mathbf{X}_{p,t}$ itself. The resulting function can be unimodal or multimodal (for details, see [1, 2]). Since the shape model is affected by noise (and consequently it cannot be defined as ideal) and observations are also affected by the environmental noise, for example, clutter in the scene and distracters, a criterion must be fixed to select representative points from the estimated function (2). One possible choice is considering a set of points such that they correspond

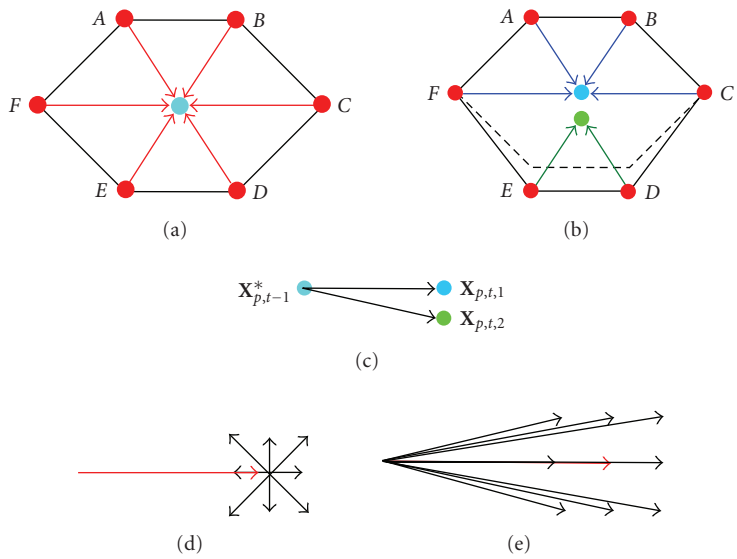


Figure 1: (a) An object model with six model corners at time $t-1$. (b) The same object at time t along with distortion of two corners "D" and "E" by one pixel in the direction of y-axis. Blue and green arrows show voting to different positions. (c) The motion vector of the reference point related to both candidates in (b). (d) The motion vector of the reference point to a voted position along with regularization. (e) Clustering nine motion hypotheses in the regularization process using a uniform kernel of radius $\sqrt{2}$.

to sufficiently high values of the estimated function (high number of votes) and they are spatially well separated. In this way, it can be shown that a set of possible alternative motion hypotheses of the object are considered corresponding to each selected point. As an example, one can have a look at Figure 1.

Figure 1(a) shows an object with six corners at time $t-1$. The corners and the reference point are shown with red color and light blue, respectively. The arrows show the position of the corners with respect to the reference point. Figure 1(b) shows the same object at time t, while two corners "D" and "E" are distorted by one pixel in the direction of y-axis. The dashed lines indicate the original figure without any change. For localizing the object, all six corners vote based on the model corners. In Figure 1(b), only six votes are shown without considering regularization. The four blue arrows show the votes of corners "A," "B," "C," and "F" for a position indicated by the light blue color. This position can be a candidate for the new reference point and is shown by $\mathbf{X}_{p,t,1}$ in Figure 1(c). Two corners "E" and "D" are voting to another position marked with a green-colored circle. This position is called $\mathbf{X}_{p,t,2}$ and it is located below the $\mathbf{X}_{p,t,1}$ with a distance of one pixel. Figure 1(c) plotted the reference point at time $t-1$ and the two candidates at time t in the same Cartesian system. Black arrows in Figure 1(c) indicate the displacement of the old reference point considering each candidate at time t to be the new reference point. These three figures make the aforementioned reasoning clearer. From the figures, it is clear that each position in the voting space corresponds either to one motion vector (if there is no regularization) or to a set of motion vectors (if there is regularization (Figures 1(d) and 1(e))). Each motion vector, in turn, corresponds to a subset of observations that are moving with the same motion. In case of regularization, these two motion vectors can be clustered together since they are very close. This is shown in Figures 1(d) and 1(e). In case of using a uniform kernel with a radius of $\sqrt{2}$ (6), all eight pixels around each position are clustered in the same cluster as the position. Such a clustering is depicted in Figures 1(d) and 1(e) where the red arrow shows the motion of the reference point at time $t-1$ to a candidate position. Figure 1(e) shows all nine motion vectors that can be clustered together. Figures 1(d) and 1(e) are equivalent. In the current work, a uniform kernel with a radius of $\sqrt{8}$ is used (therefore, 25 motion hypotheses are clustered together).

To limit the computational complexity, a limited number of candidate points, say h, are chosen (in this paper $h = 4$). If the function produced by (5) is unimodal, only the peak is selected as the only hypothesis, and hence the new object position. If it is multimodal, four peaks are selected using the method described in [1, 2]. The h points corresponding to the motion hypotheses are called *maxima* and the hypotheses set is called *the maxima set*, $H_M = \{\mathbf{X}^*_{p,t,h} \mid h = 1 \cdots 4\}$.

In the next subsection and using Figure 1, it is shown that a set of corners can be associated with each maximum h in the H_M that corresponds to observations that supported a global motion equal to the shift from $\mathbf{X}^*_{p,t-1}$ to $\mathbf{X}^*_{p,t,h}$. Therefore, the distance in the voting space between two maxima h and h' can be also interpreted as the distance between alternative hypotheses of the object motion vectors, that is, as alternative global object motion hypothesis. As a consequence, points in the H_M that are close to each other, correspond to hypotheses characterized by similar global object motion. On the contrary, points in the H_M that are far from each other correspond to hypotheses characterized by incoherent global motion hypotheses with respect to each other.

In the current paper, the point in H_M with the highest number of votes is chosen as the new object position (the *winner*). Then, other maxima in the hypotheses set are evaluated based on their distance from the winner. Any maximum that is close enough to the winner is considered as a member of the *pool of winners* W_S and the maxima that are not in the pool of winners are considered as far maxima forming the *far maxima* set $F_S = H_M - W_S$. However, having a priori knowledge about the object motion makes it is possible to choose other strategies for ranking the four hypotheses. More details can be found in [1, 2]. The next step is to classify features (hereinafter referred to as corners) based on the pool of winners and the *far maxima set*.

3.1.4. Feature classification

Now, all observations must be classified, based on their votes to the maxima, to distinguish between observations that belong to the distracter (F_S) and other observations. To do this, the corners are classified into four classes: good, suspicious, malicious, and neutral. The classes are defined in the following way.

Good corners

Good corners are those that have voted at least for one maximum in the "pool of winners" but they have not voted for any maximum in the "far maxima" set. In other words, good corners are subsets of observations that have motion hypotheses coherent with the winner maximum. This class is shown by S_G as follows:

$$S_G = \bigcup_{i=1\cdots N(W_S)} S_i - \bigcup_{j=1\cdots N(F_S)} S_j, \tag{7}$$

where S_i is the set of all corners that have voted for the ith maximum and $N(W_S)$ and $N(F_S)$ are the number of maxima in the "pool of winners" and "far maxima set" respectively.

Suspicious corners

Suspicious corners are those that have voted at least for one maximum in the "pool of winners" and they have also voted for at least one maximum in the "far maxima" set. Since corners in this set voted for pool of winners and far maxima set, they can introduce two sets of motion hypotheses. One set is coherent with the motion of the winner, while the other set of motion hypotheses is incoherent with the winner. This class is shown by S_S as follows:

$$S_S = \bigcap \left(\bigcup_{i=1\cdots N(W_S)} S_i, \bigcup_{j=1\cdots N(F_S)} S_j \right). \tag{8}$$

Malicious corners

Malicious corners are those that have voted to at least one maximum in the far maxima set, but they have not voted for any maximum in the pool of winners. Motion hypotheses corresponding to this class are completely incoherent with the object global motion. This class is formulated as follows:

$$S_M = \bigcup_{j=1\cdots N(F_S)} S_j - \bigcup_{i=1\cdots N(W_S)} S_i. \tag{9}$$

Neutral corners

Neutral corners are those that have not voted to any maximum. In other words, no decision can be made regarding the motion hypotheses of these corners. This class is shown by S_N.

These four classes are passed to the updating shape-based model module (first term in (1)).

Figure 2 shows a very simple example in which a square is tracked. The square is shown using red dots representing its corners. Figure 2(a) is the model represented by four corners $\{A1, B1, C1, D1\}$. The blue box at the center of the square indicates the reference point. Figure 2(b) shows the observations set composed by four corners. These corners are voting based on (5). Therefore, if observation A is considered as the model corner $D1$, it will vote based on the relative position of the reference point with respect to $D1$, that is, it will vote to the top left (Figure 2(b)). The arrows in Figure 2(b) show the voting procedure. In the same way, all observations vote. Figure 2(d) shows the number of votes acquired from Figure 2(b). In Figure 2(c), a triangle has been shown with its corners. The blue crosses indicate the triangle corners. In this example, the triangle is considered as a distracter whose corners are considered as a part of observations and may change the number of votes for different positions. In this case, the point "$M1$" receives five votes from $\{A, B, C, D, E\}$ (consider that due to regularization, the number of votes to "$M1$" is equal to the summation of votes to its neighbors). The relative voting space is shown in Figure 2(e). In case corner "B" is occluded, the points "$M1$" to "$M3$" will receive one vote less. The points "$M1$" to "$M3$" show three maxima. Assuming "$M1$" as the winner and as the only member of the pool of winners, $M2$ and "$M3$" are considered as far maxima: $H_M = \{M1, M2, M3\}$, $W_S = \{M1\}$, and $F_S = \{M2, M3\}$. In addition, we can define the set of corners voting for each candidate: $\text{obs}(M1) = \{A, B, C, D, E\}$, $\text{obs}(M2) = \{A, B, E, F\}$, and $\text{obs}(M3) = \{B, E, F, G\}$, where $\text{obs}(M)$ indicates the observations voting for M. Using formulas (7) to (9), observations can be classified as $S_G = \{C, D\}$, $S_S = \{A, B, E\}$, and $S_M = \{F, G\}$. In Figure 2(c), the brown cross "H" is a neutral corner ($S_N = \{H\}$) since it is not voting to any maxima.

3.2. The shape-based model

Having found the new estimated global position of the object, the shape must be estimated. This means to apply a strategy to maximize the probability of the posterior $p(\mathbf{X}_{s,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}^*_{p,t})$ where all terms in the conditional part have been fixed. Since the new position of the object $\mathbf{X}_{p,t}$ has been fixed to $\mathbf{X}^*_{p,t}$ in the previous step, the posterior can be

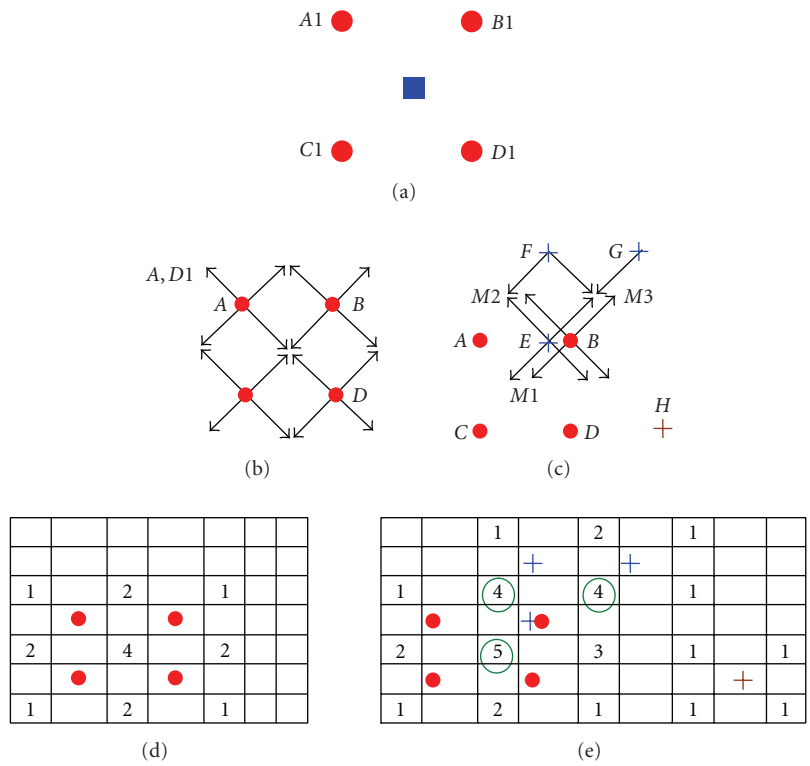


FIGURE 2: Voting and corners classification. (a) The shape model in red dots and the reference point in blue box. (b) Observations and voting in an ideal case without any distracter. (c) Voting in the presence of a distracter in blue cross. (d) The voting space related to (b). (e) The voting space related to (c) along with three maxima shown by green circles.

written as $p(\mathbf{X}_{s,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}^*_{p,t})$. With a reasoning approach similar to the one related to (2), one can write

$$p(\mathbf{X}_{s,t} \mid \mathbf{Z}_t, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}^*_{p,t}) = k' \cdot p(\mathbf{X}_{s,t} \mid \mathbf{X}^*_{s,t-1}) \cdot p(\mathbf{Z}_t \mid \mathbf{X}_{s,t}, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}^*_{p,t}), \tag{10}$$

where the first term at the right-hand side of (10) is the shape prediction model (a priori probability of the object shape) and the second term is the shape updating observation model (likelihood of the object shape).

3.2.1. The shape prediction model

Since small changes are assumed in the object shape in two successive frames, and since the motion is assumed to be independent from the shape and its local variations, it is reasonable to have the shape at time t be similar to the shape at time $t-1$. Therefore, all possible shapes at time t that can be generated from the shape at time $t-1$ with small variations form a shape subspace and they are assigned similar probabilities. If one considers the shape as generated independently by m model elements, then the probability can be written in terms of the kernel $K_{\mathrm{ls},m}$ of each model element as

$$p(\mathbf{X}_{s,t} \mid \mathbf{X}^*_{s,t-1}) = \frac{\prod_m K_{\mathrm{ls},m}(\mathbf{X}^m_{s,t}, \boldsymbol{\eta}^m_{s,t})}{\sum_{\mathbf{X}_{s,t}} \prod_m K_{\mathrm{ls},m}(\mathbf{X}^m_{s,t}, \boldsymbol{\eta}^m_{s,t})} \tag{11}$$

$\boldsymbol{\eta}^m_{s,t}$ is the set of all model elements at time $t-1$ that lies inside a rigidity kernel K_R with the radius R_R centered on $\underline{\mathbf{X}}^m_{s,t} : \boldsymbol{\eta}^m_{s,t} = \{\mathbf{X}^j_{s,t-1} : d_{m,j}(\mathbf{DX}^m_{c,t}, \mathbf{DX}^j_{c,t-1}) \le R_R\}$. The subscript ls stands for the term *"local shape."* As in (12) the local shape kernel of each shape element depends on the relation between that shape element and each single element inside the neighborhood as well as the effect of the single element on the persistency of the shape element:

$$\begin{aligned} K_{\mathrm{ls},m}(\mathbf{X}^m_{s,t}, \boldsymbol{\eta}^m_{s,t}) &= \sum_{j:\mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^m_{s,t}} K^j_{\mathrm{ls},m}(\mathbf{X}^m_{s,t}, \mathbf{X}^j_{s,t-1}) \\ &= \sum_{j:\mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^m_{s,t}} K_R(d_{m,j}(\mathbf{DX}^m_{c,t}, \mathbf{DX}^j_{c,t-1})) \cdot K^j_{P,m}(\mathbf{X}^m_{s,t}, \mathbf{X}^j_{s,t-1}). \end{aligned} \tag{12}$$

The last term in (12) allows us to define different effects on the persistency, for example, based on distance of the single element from the shape element. Here, a simple function of zero and one is used:

$$K^j_{P,m}(\mathbf{X}^m_{s,t}, \mathbf{X}^j_{s,t-1}) = \begin{cases} 1 & \text{if } (P^m_t - P^j_{t-1}) \in \{1, -1, P_{\mathrm{th}}, P_I, 0, -P^m_{t-1}\}, \\ 0 & \text{elsewhere.} \end{cases} \tag{13}$$

The set of possible values of the difference between two persistency values is computed by considering different cases (appearing, disappearing, flickering ...) that may occur for a given corner between two successive frames. For more details on how it is computed one can refer to [2].

3.2.2. The shape updating observation model

According to the shape prediction model, only a finite set (even though quite large) of possible new shapes ($\mathbf{X}_{s,t}$) can be obtained. After prediction of the shape model at time t, the shape model can be updated by an appropriate observation model that filters the finite set of possible new shapes to select one of the possible predicted new shape models ($\mathbf{X}_{s,t}$). To this end, the second term in (10) is used. To compute the probability, a function q is defined on $\mathbf{Q}$ whose domain is the coordinates of all positions inside $\mathbf{Q}$ and its range is $\{0, 1\}$. A zero value for a position (x, y) shows the lack of an observation at that position; while a one value indicates the presence of an observation at that position. The function q is defined as follows:

$$q(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \mathbf{Z}_t \\ 0 & \text{if } (x,y) \in \mathbf{Z}_t^c, \end{cases} \tag{14}$$

where $\mathbf{Z}_t^c$ is the complimentary set of $\mathbf{Z}_t$: $\mathbf{Z}_t^c = \mathbf{Q} - \mathbf{Z}_t$. Therefore, using (14) and having the fact that the observations in the observations set are independent from each other, the second probability term in (10) can be written as a product of two terms:

$$\begin{aligned} p(\mathbf{Z}_t \mid \mathbf{X}_{s,t}, \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}^*_{p,t}) \\ = \prod_{(x,y)\in\mathbf{Z}_t} p(q(x,y) = 1 \mid \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}_{s,t}, \mathbf{X}^*_{p,t}) \\ \cdot \prod_{(x,y)\in\mathbf{Z}_t^c} p(q(x,y) = 0 \mid \mathbf{X}^*_{p,t-1}, \mathbf{X}^*_{s,t-1}, \mathbf{X}_{s,t}, \mathbf{X}^*_{p,t}). \end{aligned} \tag{15}$$

Based on the presence or absence of a given model corner in two successive frames and based on its persistency value, different cases for that model corner can be investigated in two successive frames. Investigating different cases, the following rule is derived that maximizes the probability value in (15) [2]:

$$P_t^n = \begin{cases} P_{t-1}^j + 1 & \text{if } \exists j : \mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^n_{s,t},\ P^j_{t-1} \geq P_{\text{th}},\ q(x_n, y_n) = 1 \\ & \text{(the corner exists in both frames)}, \\ P_{t-1}^j - 1 & \text{if } \exists j : \mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^n_{s,t},\ P^j_{t-1} > P_{\text{th}},\ q(x_n, y_n) = 0 \\ & \text{(the corner disappears)}, \\ 0 & \text{if } \exists j : \mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^n_{s,t}, P^j_{t-1} = P_{\text{th}},\ q(x_n, y_n) = 0 \\ & \text{(the corner disappears)}, \\ P_I & \text{if } \nexists j : \mathbf{X}^j_{s,t-1} \in \boldsymbol{\eta}^n_{s,t},\ q(x_n, y_n) = 1 \\ & \text{(a new corner appears)}. \end{cases} \tag{16}$$

To implement model updating, (16) is implemented considering also the four classes of corners (see Section 3.1.4). To this end, the corners in the malicious class are discarded. The corners in the good and suspicious classes are fed to formula (16). The neutral corners can be treated differently. They may be fed to (16). This can be done when all hypotheses belong to the pool of winners, that is, when no distracter is available. If this is not the case, the neutral corners are also discarded. Although some observations are discarded (malicious corners), the compliancy to the Bayesian framework is achieved through the following:

(i) adaptive shape noise (e.g., occluder, clutter, distracter) model estimation;

(ii) filtering observations $\mathbf{Z}_t$ to produce a reduced observation set $\mathbf{Z}'_t$;

(iii) substitute $\mathbf{Z}'_t$ in (15) to compute an alternative solution $\mathbf{X}'_{s,t}$.

The above-mentioned procedure simply says that discarded observations are noise.

In the first row of (16), there may be more than one corner in the neighborhood of a given corner ($\boldsymbol{\eta}^n_{s,t}$). In this case, the closest one to the given corner is chosen, see [1, 2] for more details on updating.

4. COLLABORATIVE TRACKING

Independent trackers are prone to merge error and labeling error in multiple target applications. While it is a common sense that a corner in the scene can be generated by only one object and can therefore participate in the position estimation and shape updating of only one tracker, this rule is systematically violated when multiple independent trackers come into proximity. In this case, in fact, the same corners are used during the evaluation of (2) and (10) with all problems described in the related work section. To avoid these problems, an algorithm that allows the collaboration of trackers and that exploits feature classification information is developed. Using this algorithm, when two or more trackers come to proximity, they start to collaborate both during the position and the shape estimation.

4.1. General theory of collaborative tracking

In multiple object tracking scenarios, the goal of the tracking algorithm is to estimate the joint state of all tracked objects $[\mathbf{X}^1_{p,t}, \mathbf{X}^1_{s,t}, \ldots, \mathbf{X}^G_{p,t}, \mathbf{X}^G_{s,t}]$, where G is the number of tracked objects. If objects observations are independent, it will be possible to factor the distributions and to update each tracker separately from others.

In case of dependent observations, their assignments have to be estimated considering the past shapes and positions of interacting trackers. Considering that not all trackers interact (far objects do not share observations), it is possible to simplify the tracking process by factoring the joint posterior in dynamic collaborative sets. The trackers should be divided into sets considering their interactions: one set for each group of interacting targets.

To do this, the overlap between all trackers is evaluated by checking if there is a spatial overlap between shapes of trackers at time $t-1$.

The trackers are divided into J sets such that objects associated to trackers of each set interact with each other within the same set (intraset interaction) but they do not overlap any tracker of any other set (there is no interset interaction).

Since there is no interset interaction, observations of each tracker in a cluster can be assigned conditioning only on trackers in the same set. Therefore, it is possible to factor the joint posterior into the product of some terms each of which assigned to one set:

$$\begin{aligned} p(\mathbf{X}_{p,t}^1, \mathbf{X}_{s,t}^1, \ldots, \mathbf{X}_{p,t}^G, \mathbf{X}_{s,t}^G \mid \mathbf{Z}_t^1, \ldots, \mathbf{Z}_t^G, \\ \mathbf{X}_{p,t-1}^{*1}, \mathbf{X}_{s,t-1}^{*1}, \ldots, \mathbf{X}_{p,t-1}^{*G}, \mathbf{X}_{s,t-1}^{*G}) \\ = \prod_{j=1}^{J} p\left(\mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{s,t}^{N_t^j} \mid \mathbf{Z}_t^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}, \mathbf{X}_{s,t-1}^{*N_t^j}\right), \end{aligned} \tag{17}$$

where J is the number of collaborative sets and $\mathbf{X}_{p,t}^{N_t^j}$, $\mathbf{X}_{s,t}^{N_t^j}$ and $\mathbf{Z}_t^{N_t^j}$ are the states and observations of all trackers in the set N_t^j, respectively. In this way, there is no necessity to create a joint-state space with all trackers, but only J spaces. For each set, the solution to the tracking problem is estimated by calculating the joint state in that set that maximizes the posterior of the same collaborative set.

4.2. Collaborative position estimation

When an overlap between the trackers is reported, they are assigned to the same set N_t^j. While the a priori position prediction is done independently for each tracker in the same set (3), the likelihood calculation, that is not factorable, is done in a collaborative way.

The union of observations of trackers in the collaborative set $\mathbf{Z}_t^{N_t^j}$ is considered as generated by L trackers in the set. Considering that during an occlusion event, there is always an object that is more visible than the others (the occluder), with the aim of maximizing (17), it is possible to factor the likelihood in the following way:

$$\begin{aligned} p\left(\mathbf{Z}_t^{N_t^j} \mid \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}\right) \\ = p\left(\Xi \mid \mathbf{Z}_t^{N_t^j} \setminus \Xi, \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}\right) \\ \cdot p\left(\mathbf{Z}_t^{N_t^j} \setminus \Xi \mid \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}\right), \end{aligned} \tag{18}$$

where the observations are divided into two sets: $\Xi \subset \mathbf{Z}_t^{N_t^j}$ and the remaining observations $\mathbf{Z}_t^{N_t^j} \setminus \Xi$. To maximize (18), it is possible to proceed by separately (and suboptimally) finding a solution to the two terms assuming that the product of the two partial distributions will give rise to a maximum in the global distribution. If the lth object is perfectly visible, and if Ξ is chosen as $\mathbf{Z}_t^l$, the maximum will be generated only by observations of the lth object. Therefore, one can write

$$\begin{aligned} \max p\left(\Xi \mid \mathbf{Z}_t^{N_t^j} \setminus \Xi, \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}\right) \\ = \max p(\mathbf{Z}_t^l \mid \mathbf{X}_{s,t-1}^{*l}, \mathbf{X}_{p,t}^l, \mathbf{X}_{p,t-1}^{*l}). \end{aligned} \tag{19}$$

Assuming that the tracker under analysis is associated to the most visible tracker, it is possible to use the algorithm described in Section 3 to estimate its position using all observations $\mathbf{Z}_t^{N_t^j}$.

It is possible to state that the position of the winner maximum estimated using all observations $\mathbf{Z}_t^{N_t^j}$ will be in the same position as if it were estimated using $\mathbf{Z}_t^l$. This is true because if all observations of the lth tracker are visible, $p(\mathbf{Z}_t^{N_t^j} | \mathbf{X}_{s,t-1}^{*l}, \mathbf{X}_{p,t}^l, \mathbf{X}_{p,t-1}^{*l})$ will have one peak in $\mathbf{X}_{p,t}^{*l}$ and some other peaks in correspondence of some positions that correspond to groups of observations that are similar to $\mathbf{X}_{s,t-1}^{*l}$. However, using motion information as well, it is possible to filter existing peaks which do not correspond to the true position of the object. Using the selected winner maximum and the classification information, one can estimate the set of observations Ξ. To this end, only S_G (7) is considered as Ξ. Corners that belong to S_S (8) and S_M (9) have voted for the far maxima as well. Since in an interaction, far maxima can be generated by the presence of some other object, these corners may belong to other objects. Considering that the assignment of the corners belonging to the S_S is not certain (considering the nature of the set), the corners belonging to this set are stored together with the neutral corners S_N for an assignment revision in the shape-estimation step.

So far, it has been shown how to estimate the position of the most visible object and the corners belonging to it, assuming that the most visible object is known. However, the ID, position, and observations of the most visible object are all unknown and they should be estimated together. To do this, to find the tracker that maximizes the first term of (18), the single tracking algorithm is applied to all trackers in the collaborative set to select a winner maximum for each tracker in the set using all observations associated to the ser $\mathbf{Z}_t^{N_t^j}$.

For each tracker l, the ratio $Q(l)$ between the number of elements in its S_G and in its shape model $\mathbf{X}_{t-1,s}^{*l}$ is calculated. A value near 1 means that all model points have received a vote, and hence there is full visibility, while a value near 0 means full occlusion. The tracker with the highest value of $Q(l)$ is considered as the most visible one and its ID is assigned to $O(1)$ (a vector that keeps the order of estimation). Then, using the procedure described in Section 3, its position is estimated and is considered as the position of its winner maximum. In a similar manner, its observations $\mathbf{Z}_t^{O(1)}$ are considered as the corners belonging to the set Ξ.

To maximize the second term of (18), it is possible to proceed in an iterative way. The remaining observations are the observations that remain in the scene when the evidence that certainly belongs to $O(1)$ is removed from the scene. Since there is no evidence of the tracker $O(1)$, by defining

$\mathbf{Z}_t^{N_t^j \setminus O(1)}$ as $\mathbf{Z}_t^{N_t^j} \setminus \Xi$, it is possible to state that

$$\max p\left(\mathbf{Z}_t^{N_t^j \setminus O(1)} \mid \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}\right) = \max p\left(\mathbf{Z}_t^{N_t^j \setminus O(1)} \mid \mathbf{X}_{s,t-1}^{*N_t^j \setminus O(1)}, \mathbf{X}_{p,t}^{N_t^j \setminus O(1)}, \mathbf{X}_{p,t-1}^{*N_t^j \setminus O(1)}\right). \tag{20}$$

Now, one can sequentially estimate the next tracker by iterating (18).

Therefore, it is possible to proceed greedily with the estimation of all trackers in the set. To this end, the order of the estimation, the position of the desired object, and corners assignment are estimated at the same time. The objects that are more visible are estimated at the beginning and their observations are removed from the scene. During shape estimation, corner assignment will be revised using feature classification information and the models of all objects will be updated accordingly.

4.3. Collaborative shape estimation

After estimation of the objects positions in the collaborative set (here it is indicated with $\mathbf{X}_{p,t}^{*N_t^j}$), their shapes should be estimated. The shape model of an object cannot be estimated separately from the other objects in the set, because each object may occlude or be occluded by the others. For this reason, the joint global shape distribution is factored in two parts, the first one predicts the shape model, and the second term refines the estimation using the observation information. With the same reasoning that led to (10), it is possible to write

$$p\left(\mathbf{X}_{s,t}^{N_t^j} \mid \mathbf{Z}_t^{N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}, \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{*N_t^j}\right) = k \cdot p\left(\mathbf{X}_{s,t}^{N_t^j} \mid \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{*N_t^j}\right) \cdot p\left(\mathbf{Z}_t^{N_t^j} \mid \mathbf{X}_{s,t}^{N_t^j}, \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{*N_t^j}\right), \tag{21}$$

where k is a normalization constant. The dependency of $\mathbf{X}_{s,t}^{N_t^j}$ on the current and past positions means that the a priori estimation of the shape model should take into account the relative positions of the tracked object on the image plane.

4.3.1. A priori collaborative shape estimation

The a priori joint shape model is similar to the single object model. The difference with the single object case is that in the joint shape estimation model, points of different trackers that share the same position on the image plane cannot increase their persistency at the same time. In this way, since the increment of persistency of a model point is strictly related to the presence of a corner in the image plane, the common sense stating that each corner can belong only to one object is implemented [4].

The same position on the image plane of a model point corresponds to different relative positions in the reference system of each tracker; that is, it depends on the global positions of the trackers at time t, $\mathbf{X}_{p,t}^{*N_t^j}$. For a tracker j, the lth model point has the absolute position $(x_m^j, y_m^j) = (x_{\text{ref}}^j + dx_m^j, y_{\text{ref}}^j + dy_m^j)$. This consideration is easily understood from Figure 3 where a model point is on the left shown in its absolute position while the trackers have been centered on their estimated global positions. On the right side of Figure 3, each tracker is considered by itself with the same model point highlighted in the local reference system.

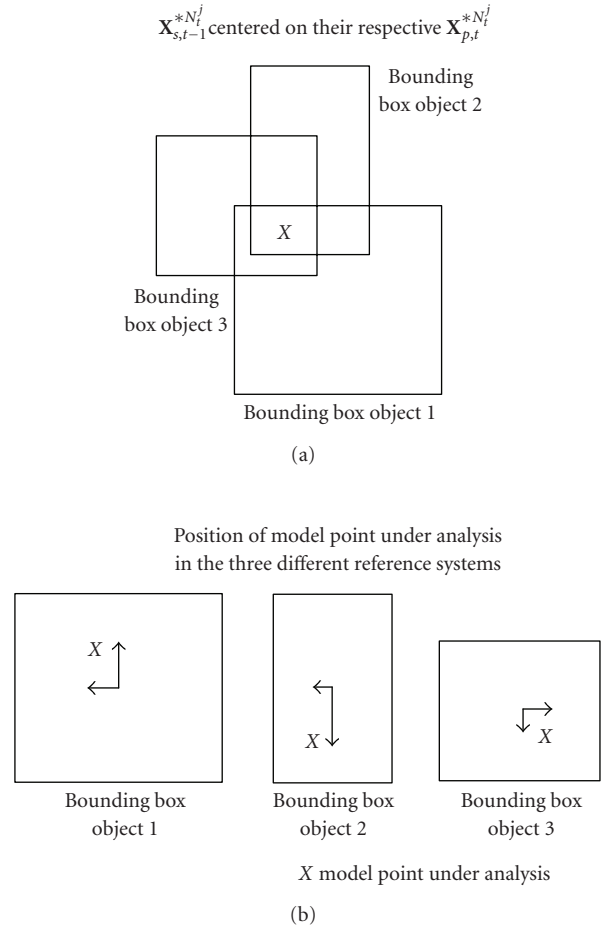


Figure 3: Example of the different reference systems in which it is possible to express the coordinates of a model point. (a) three model points of three different trackers share the same absolute position (x_m, y_m) and hence they belong to the same set C^m. (b) the three model points are expressed in the reference system of each tracker.

The framework derived for the single object shape estimation is here extended with the aim of assigning a zero probability to configurations in which multiple model points that lie on the same absolute position have an increase of persistency.

Given an absolute position (x_m, y_m), it is possible to define the set C^m which contains all the model points of the trackers in the collaborative set that are projected with respect to their relative position on the same position (x_m, y_m) (see Figure 3).

Considering all the possible absolute positions (the positions that are covered by at least one bounding box of the

trackers in N_t^j), it is possible to define the following set that contains all the model points that share the same absolute position with at least another model point of another tracker,

$$I = \{C^i : \text{card}\ (C^i) > 1\}. \tag{22}$$

In Figure 3, it is possible to visualize all the model points that are part of I as the model points that lie in the intersection of at least two bounding boxes. With this definition, it is possible to factor the a priori shape probability density in two different terms as follows:

(1) a term that takes care of the model points that are in a zone where there are not model points of other trackers (model points that do not belong to I);

(2) a term that takes care of the model points that belong to the zones where the same absolute position corresponds to model points of different trackers (model points that belong to I).

This factorization can be expressed in the following way:

$$\begin{aligned} p\left(\mathbf{X}_{s,t}^{N_t^j} \mid \mathbf{X}_{s,t-1}^{*N_t^j}, \mathbf{X}_{p,t-1}^{*N_t^j}, \mathbf{X}_{p,t}^{*N_t^j}\right) \\ = k\left[\prod_{m\notin I} K_{\text{ls},m}(\mathbf{X}_{s,t}^m, \boldsymbol{\eta}_{s,t}^m)\right] \\ \times\left[\prod_{m\in I} K_{\text{ex}}(\mathbf{X}_{s,t}^{C^m}, \boldsymbol{\eta}_{s,t}^{C^m}) \prod_{i\in C^m} K_{\text{ls},m}(\mathbf{X}_{s,t}^{C^m(i)}, \boldsymbol{\eta}_{s,t}^{C^m(i)})\right], \end{aligned} \tag{23}$$

where k is a normalization constant. The first factor is related to the first bullet. It is the same as in the noncollaborative case. The model points that lie in a zone where there is no collaboration in fact follow the same rules of the single tracking methodology.

The second factor is instead related to the second bullet. This term is composed by two subterms. The rightmost product, by factoring the probabilities of model points belonging to the same C^m using the same kernel as in (12), considers each model point independently from the others even if they lie on the same absolute position. The first subterm $K_{\text{ex}}(\mathbf{X}_{s,t}^{C^m}, \boldsymbol{\eta}_{s,t}^{C^m})$, named the exclusion kernel, is instead in charge of setting the probability of the whole configuration involving the model points in C^m to zero if the updating of the model points in C^m are violating the "exclusion rule" [4].

The exclusion kernel is defined in the following way:

$$\begin{aligned} K_{\text{ex}}(\mathbf{X}_{s,t}^{C^m}, \boldsymbol{\eta}_{s,t}^{C^m}) \\ = \begin{cases} 1 & \text{if } \forall P_{t-1}^l \in \boldsymbol{\eta}_{s,t}^{C^m(i)},\ (P_t^{C^m(i)} - P_{t-1}^j) \in \{1, P_w\} \\ & \text{for no more than one model point } i \in C^m, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{24}$$

The kernel in (24) implements the exclusion principle by not allowing configurations in which there is an increase in persistency for more than one model point belonging to the same absolute position.

4.3.2. Collaborative shape updating observation model with feature classification

The shape updating likelihood, once the a priori shape estimation has been carried on in a joint way, is similar to the noncollaborative case. Since the exclusion principle has been used in the a priori shape estimation, and since each tracker has the list of its own features available, it would be possible to simplify the rightmost term in (21) by using directly (15) for each tracker. As already stated in the introduction, in fact, the impossibility in segmenting the observations is the cause of the dependence of the trackers; at this stage, instead, the feature segmentation has already been carried on. It is however possible to exploit the feature classification information in a collaborative way to refine the shape classification and have a better shape estimation process. This refinement is possible because of the joint nature of the right term in (21) and it would not be possible in an independent case.

Each tracker i belonging to N_t^j has, at this stage, already classified its observations in four sets:

$$\mathbf{Z}_t^{N_t^j(i)} = \left\{S_G^{N_t^j(i)}, S_S^{N_t^j(i)}, S_M^{N_t^j(i)}, S_N^{N_t^j(i)}\right\}. \tag{25}$$

Since a single object tracker does not have a complete understanding of the scene, the proposed method lets the information about feature classification be shared between the trackers for a better classification of features that belong to the set N_t^j. As an example to motivate this refinement, a feature could be seen as a part of S_N by one tracker (say tracker 1) and as a part of S_G by another tracker (say tracker 2). This means that the feature under analysis is classified as "new" by tracker 1 even if it is generated, with a high confidence, by the object tracked by the second tracker (see, e.g., Figure 4). This situation is by common sense due to the fact that, when two trackers come into proximity, the first tracker sees the feature belonging to the second tracker as a new feature.

If two independent trackers were instantiated, in this case, tracker 1 would erroneously insert the feature into its model. By sharing information between the trackers, it is instead possible to recognize this situation and prevent that the feature is added by tracker 1.

To solve this problem, the list of classified information is shared by the trackers belonging to the same set. The following two rules are implemented.

(i) If a feature is classified as good (belonging to S_G) for a tracker, it is removed from any S_S or S_N of other trackers.

(ii) If a feature is classified as suspicious (belonging to S_S) for a tracker, it is removed from any S_N of other trackers.

By implementing these rules, it is possible to remove the features that belong to other objects with a high confidence from the lists of classified corners of each tracker. Therefore, for each tracker, the modified sets S_S' and S_N' are obtained. The S_G and S_M will be instead unchanged (see Figure 4(e)).

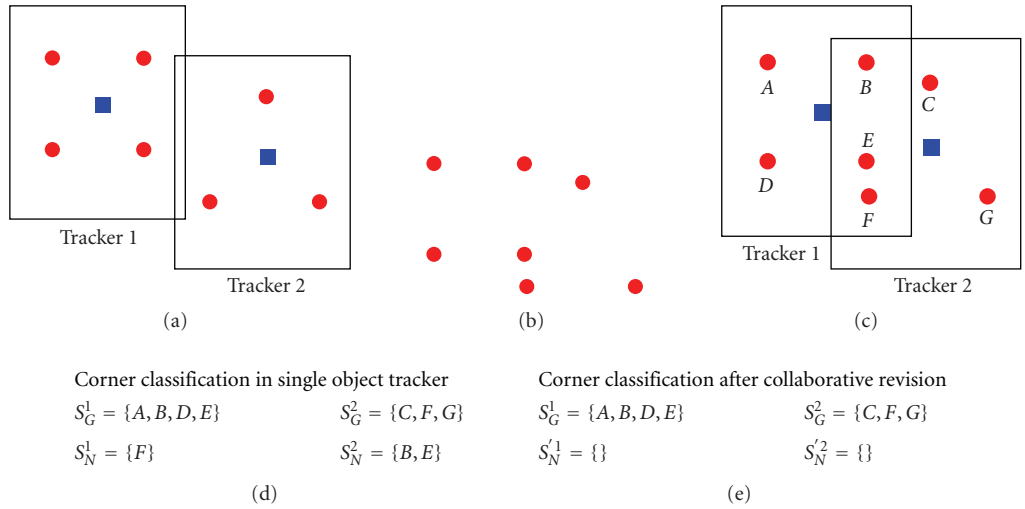


Figure 4: Collaborative corners revision. (a) The shape models in red dots and the reference points in blue boxes of two trackers at time $t-1$. (b) Observations at time t of the two interacting objects. (c) Positions (correctly) estimated by the position updating step. The observations of tracker 1 are in the bounding box of tracker 2 and vice versa. (d) Corners classification by each tracker. (e) The corners are revised by the collaborative classification rules.

After this refinement, each tracker will form the set of observations called $\mathbf{Z}_t'^{N_t^j(i)}$ as the union of S_G, S_S' which will be used in the shape estimation and S_N'. These reduced sets of observations are therefore used, instead of the $\mathbf{Z}_t^{N_t^j(i)}$ in (15).

The joint effect of the collaborative a priori shape estimation with exclusion principle (that prevents that multiple model points of different trackers in the same absolute position are increased in persistency at the same time) and the corner classification refinement will, as it will be clear in the experimental result part, greatly improve the shape estimation process in case of interaction of the trackers.

4.4. Collaborative tracking discussion

Using the proposed framework, an algorithm for the collaborative estimation of the states of interacting targets has been developed. In this section, the algorithm will be discussed through an example.

$O(0) = \varnothing$
for $i = 0$ to $L-1$
  $R = N_t^j \setminus [O(0) \cdots O(i)]$
  for $l = 1$ to $L - i$
    Winner maximum estimation for tracker $R(l)$
    Corner classification for tracker $S(l)$
    $Q(l) = \text{size}(S_G^l)/\text{size}(\underline{X}_{s,t-1}^l)$
  end
  $O(i) = R\left(\underset{k}{\text{argmax}}\, Q(k)\right)$
  Assign the winner maximum position to $\mathbf{X}_{p,t}^{O(i)}$
  Assign the corners in S_G, S_S, and S_N of $O(i)$ to the set of observation of $O(i)$
  Remove S_G of $O(i)$ from the scene
end

Algorithm 1: Position estimation in collaborative tracking.

4.4.1. Position estimation

During trackers interaction, the collaborative tracking algorithm uses the single tracking maximum selection strategy as a base for the estimation of the winner maxima of the trackers.

The tracking procedure which is an implementation of the sequential procedure described in Section 4.2 is described in Algorithm 1. The collaborative tacking procedure will be explained discussing a step of the algorithm using a sequence taken from the PETS2001 dataset. Figure 5(a) shows the tracking problem. There is a man that is partially occluded by a van. At the first step, as explained in Table 1, the single object tracking algorithm is used for both the van and the man as if they were alone in the scene. The likelihood is shown in Figure 6. In Figures 6 and 7, the winner maximum is plotted as a square, while the far maxima (that are a symptom of multimodality) are plotted as circles. To maintain the figures clear, the close maxima that are near the winner maximum are not plotted because they do not give information about the multimodality of the distribution. Since always 4 maxima are extracted, considering that one of them is the winner, in the figures where less than 4 maxima are plotted, it should be considered that k (where $k = 3$-number of far maxima) near maxima are extracted but not plotted. As it is possible to see, while the voting space of the van in Figure 6(b) is mainly monomodal (only one far maximum is extracted), the voting space in Figure 6(a) that corresponds to the occluded man is multimodal, because also the corners belonging to the van participate in the voting.

The $Q(l)$ for each object is evaluated and the van is correctly selected as the first object to be estimated and its

(a) (b) (c)

Figure 5: (a) Problem formulation. (b) Van's good corners at first step. (c) Man's good corners at first step.

Table 1: Position estimation in collaborative tracking.

	Sequence 1 108 frames resolution 720 × 576	Sequence 2 85 frames resolution 320 × 240	Sequence 3 101 frames resolution 768 × 576
Collaborative with feature classification	Successful	Successful	Successful
Noncollaborative with feature classification	Fails at the interaction between tracked targets	Fails for model corruption at frame 80	Fails at the interaction between tracked targets
Collaborative without feature classification	Successful	Fails	Fails after few frames

ID is assigned to $O(1)$. Since the van is perfectly visible, it is possible to estimate its position $\mathbf{X}_{p,t}^{O(1)}$ as if it were alone in the scene; the corners belonging to S_G, S_S, and S_N are assigned as the observations that belong to the tracker $O(1)$ (see Figure 5(b)); the corners belonging to S_G are removed from the scene, and the procedure is iterated. In this way only the corners that do not belong to the van (Figure 5(c)) are used for the estimation of the position of the man. As it is possible to see from Figure 7(a) the voting space of the man, after the corners of the van have been removed, is monomodal (the number of far maxima is decreased from 3 as in Figure 6(a) to 1) and its maximum corresponds to the position of man in the image.

The collaborative shape estimation is a greedy method that solves at the same time the position estimation and the observation assignment. As all the sequential methods, it is in theory prone to accumulation of errors (i.e., a wrong estimation of one tracker can generate an unrecoverable failure in the tracking process of all the trackers that are estimated after it). Our method is however quite robust.

The proposed method can make essentially two kinds of errors:

(1) an error in the estimation order;

(2) an error in the estimated position of one or more trackers.

An error of kind 1 would damage the shape model but, due to the presence in the shape model of points with high persistency, the model would resist for some frames to this kind of error. An error of kind 2 generates a displacement of the bounding box that can, in the shape updating process, damage the model. Since the search space (defined as the zone where the observations are extracted) in the collaborative modality is the union of the search regions of the single trackers, in case of collaboration, a displacement error can be recovered with more ease since the search region is larger.

4.4.2. *Shape estimation*

The shape estimation is realized in two steps: at first the a priori shape estimation with exclusion principle is realized, and then the estimation is corrected by using the observations. The a priori shape estimation does not allow an increase of persistency in more than a model point if they lie on the same absolute position. After this joint shape a priori estimation, the assignment of features is revised using the feature classification information in a collaborative way.

As a first step, for each tracker, a set containing the corners form S_G, S_S, and S_N is created. Each tracker has in S_G the corners that have voted only for its winner maximum and that were removed from the scene, and in S_S, the corners that have voted both for the winner maxima and one of the far maxima. As from the single tracker perspective, it is not possible to decide if these corners have voted for the winner maxima because of a false match or because they really belong to that tracker, the trackers share their information about the corners and decide the assignment of each of them.

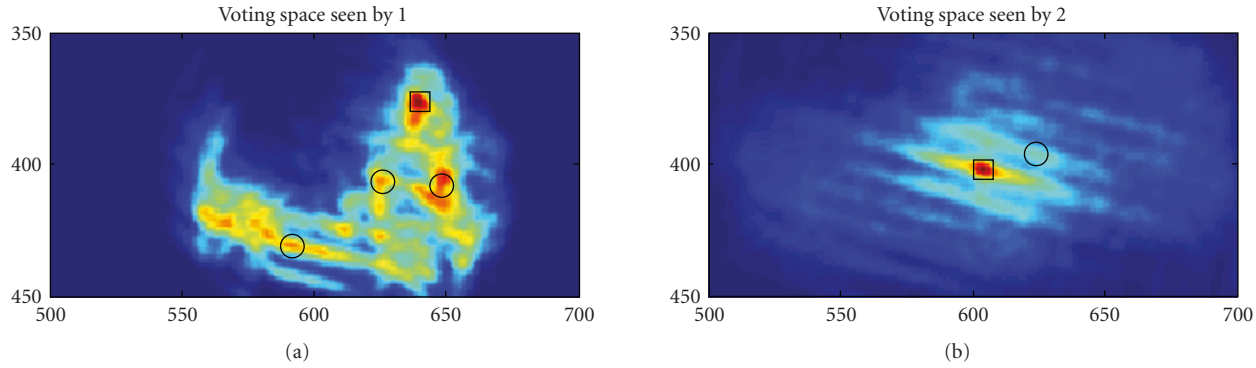


FIGURE 6: Likelihood estimation at first step. (a) $V(\cdot)$ seen by the man's tracker. (b) $V(\cdot)$ seen by the van's tracker. The winner maximum is plotted as a square and the far maxima are plotted as circles.

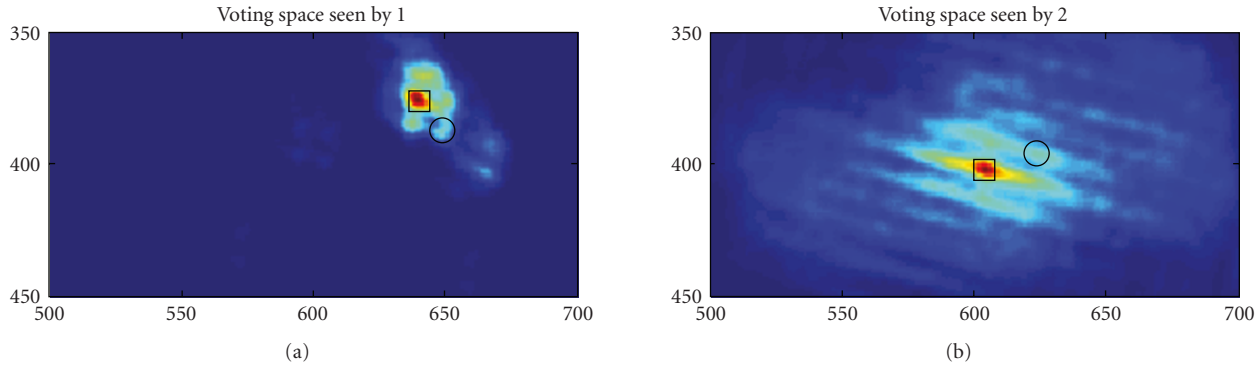


FIGURE 7: Likelihood estimation at second step. (a) $V(\cdot)$ seen by the man's tracker. (b) $V(\cdot)$ seen by the van's tracker. The winner maximum is plotted as a square and the far maxima are plotted as circles.

FIGURE 8: Corners updated in the van's and person's models.

The rules defined in Section 4.3.2 are implemented and the corners are assigned accordingly. The results of applying this method to the frame in Figure 5(a) are reported in Figure 8 where the model points that had an increase in persistence, considering both the shape a priori estimation and the likelihood calculation are shown. As it is possible to see from the figure that each tracker was updated correctly (no model points were added or increased in persistence for the man in the zone of the van and vice versa) demonstrating a correct feature classification.

5. EXPERIMENTAL RESULTS

The proposed method has been evaluated on a number of different sequences.

This section illustrates and evaluates the performances by discussing some examples that show the benefits of the collaborative approach and by proposing some qualitative and quantitative results. The complete sequences presented in this paper are available at our website [17].

As a methodological approach for the comparison, considering that this paper focuses on the classification of the features in the scene and on their use for collaborative tracking, mainly results related to occlusion situations will be presented.

5.1. *Comparison with single-tracking methodology*

At first, the results of the new collaborative approach have been compared to the results obtained using the single tracking methodology and to an approach of collaborative tracking that does not use classification information [18].

The first example (Sequence 1) is a difficult occlusion scene taken from the PETS2006 dataset where three objects

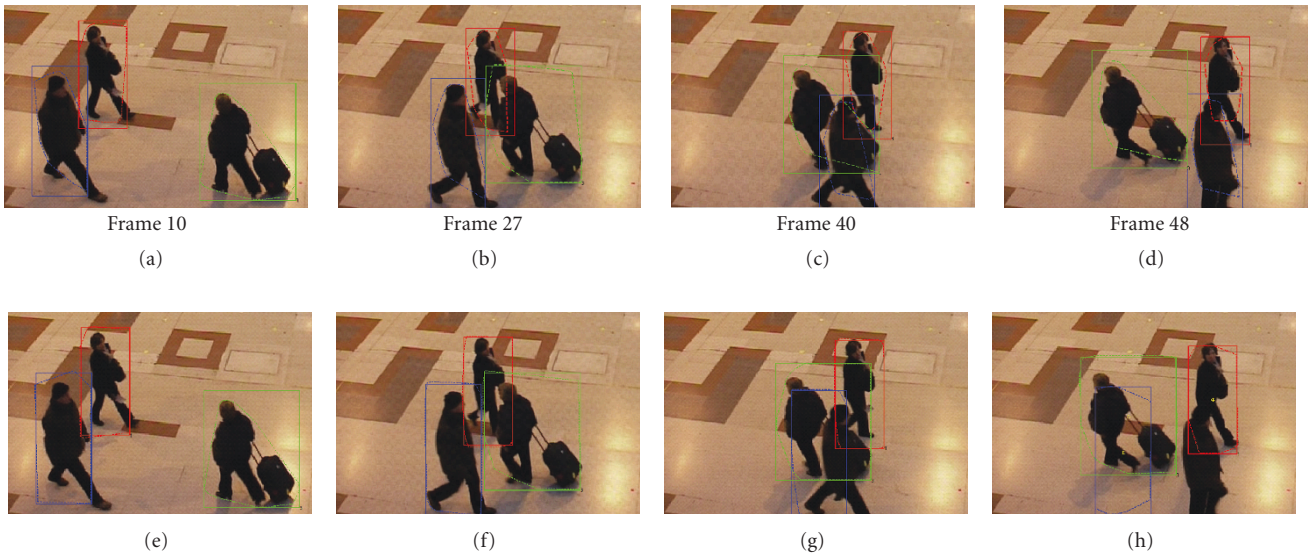


Figure 9: Sequence 1. Tracking results. Upper row: collaborative approach. Lower row: noncollaborative approach.

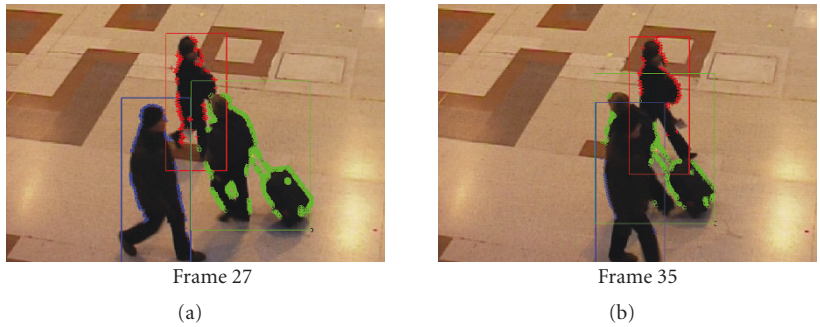


Figure 10: Sequence 1. Shape-updating results during collaboration.

interact with severe occlusion. In the sequence, two persons coming from the left of the scene interact constantly with partial occlusion; another person with a trolley coming from the right walks between the two persons occluding one of them and being occluded by the other. The persons wear dark clothes, and hence it is difficult to extract corners when there is an overlap between the silhouettes. In Figure 9, the results obtained using three independent trackers (bottom row) and those obtained using the proposed collaborative method (upper row) are compared.

As it is possible to see, one of the independent trackers looses track because it uses for shape-model updating the corners belonging to the other tracker. In the figure, the convex hull of $\mathbf{X}_{s,t}$ centred on $\mathbf{X}_{p,t}$ is plotted using a dashed line while the bounding box is plotted using a solid line. As it is possible to see by comparing the estimated shapes, the shape estimation using the collaborative methodology is more accurate than the independent one.

In Figure 10, the updated corners are plotted on top of the image to show which corners are updated and by which tracker (the background corners are not plotted for ease of visualization). As it is possible to see, each model is updated using only the corners that belong to the tracked object and not to other objects.

At frame 35, it is possible to see that the shape model of the man coming from the right is correctly updated at its left and right sides, while in the centre, due to the presence of the foreground person, the algorithm chooses not to update the model.

The next example (Sequence 2) is a sequence taken from a famous soccer sequence in which the tracked objects are involved in constant self occlusion. The quality of the images is low due to compression artefacts. In Figure 11, some frames of the sequence are displayed. As it is possible to see, the tracking process is successful and the shape is updated correctly. Figure 12(a) contains the results obtained with the proposed approach while Figure 12(b) contains the results of the noncollaborative approach. As it is possible to see, the convex hull of $\mathbf{X}_{s,t}$ is accurate in case of collaboration. If the trackers do not share information, they include in their model also corners of other objects (see the central object in Figure 12(b)) and fail in few frames due to model corruption.

In Figure 13, a comparison between the proposed approach (Figure 13 central column) and a collaborative

Figure 11: Sequence 2. Collaborative approach with feature classification results.

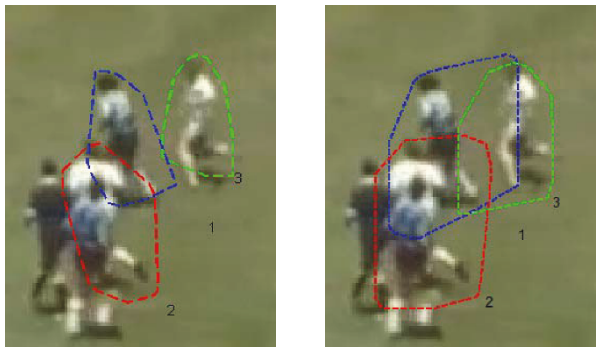


Figure 12: Sequence 2. Comparison between (a) collaborative and (b) noncollaborative approaches.

approach that does not use feature classification (Figure 13 right column) is reported. As it is possible to see, the feature classification allows the rejection of clutter (some soccer players are not tracked and are therefore to be considered clutter). Using a tracker without feature classification, corners that belong to clutter are used for shape estimation; and for this reason, tracking will fail in few frames (third row of Figure 13).

Another example (Sequence 3) is the sequence that was used to discuss the collaborative position and shape updating in Section 4.4. In Figure 14 left column, the results obtained using the noncollaborative approach are reported. The noncollaborative approach fails because the corners of the van are included in the model of the person and its model is corrupted. In the collaborative approach, the shape is correctly estimated as shown in Figure 14 right column.

In Table 1, a summary of the results is reported.

5.2. *Tracking results on long sequences*

The proposed approach has been tested on two long sequences from the PETS2006 dataset for a total of 5071 frames. The trackers were initialized and uninitialized manually. The results are summarized in Table 2 where long-term occlusion means an interaction of more than 40 frames. As it is possible to see, there are only two errors (which can be

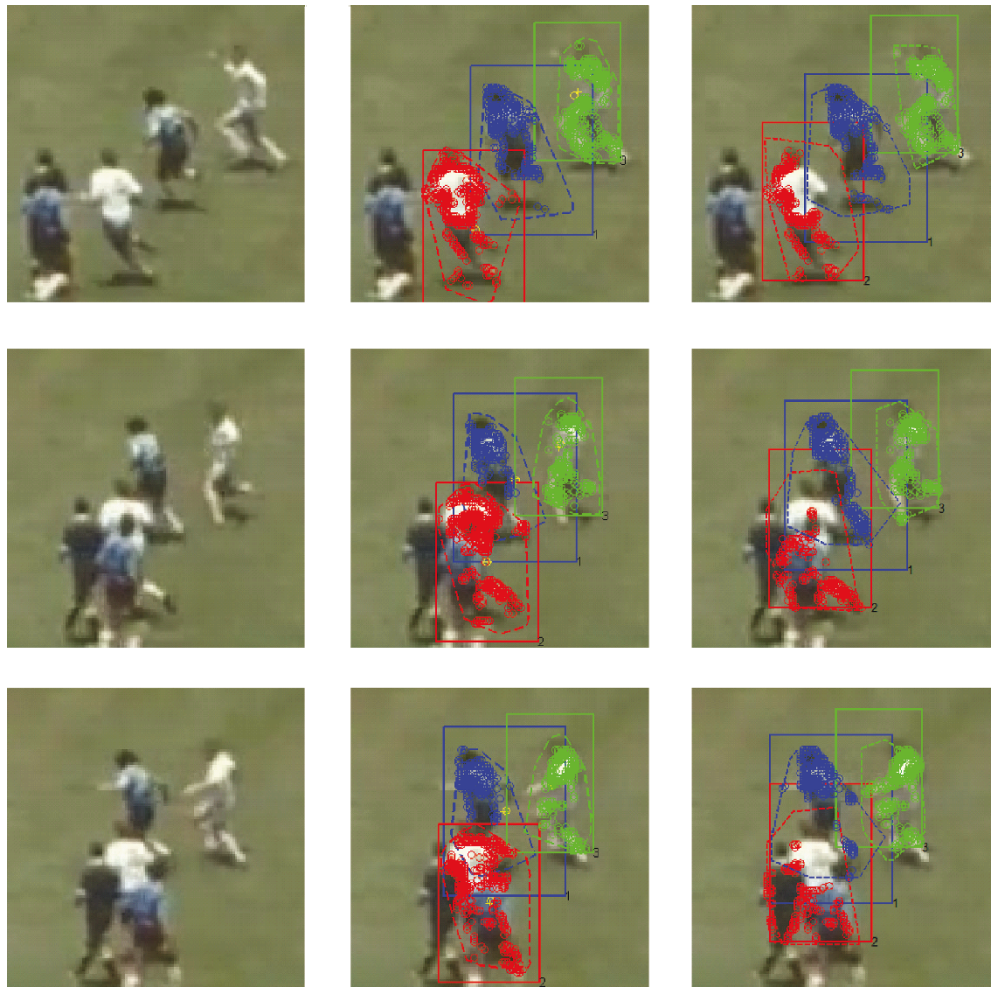


Figure 13: Sequence 2. Comparison between collaborative approach with feature classification (central column) and collaborative approach without feature classification (right column).

considered as one error since the two trackers which failed were collaborating) that occur after 500 frames from the initialization of the trackers. During these 500 frames, the trackers experienced a constant interaction between them and with another tracker in a cluttered background. The low-error rate demonstrates the capabilities of the algorithm also on long sequences. Sequences 4 and 5 are available at our website [17].

5.3. Quantitative-shape estimation evaluation

To discuss the shape-updating process and the benefits of collaboration from a quantitative point of view, some results that are related to Sequence 2 are provided in Figures 15 and 16. A particularly interesting measure of the benefits of the collaborative approach with feature classification is presented in Figure 15. The convex hull of the target labeled as 1 in Figure 11 has been manually extracted. By counting the number of model points that, due to the updating process, have an increased persistence and that are outside this manually extracted convex hull and which therefore belong to clutter or other objects, it is possible to have a measure of the correctness of the shape update process. From the graph, it is clear that from frames 3 to 10 and particularly after frame 55 (the heavy occlusion of Figure 12), many more model points are updated or added outside the target's convex hull by the noncollaborative approach. This is due to the fact the algorithm inserted model points using the evidence of other objects. This errors lead to a failure after few frames (for this reason, the noncollaborative approach graph ends at frame number 80).

In Figure 16, the number of updated, decreased in persistence, removed, and added model points in collaborative (left column) and noncollaborative approach (right column) for targets labeled as 1 (first row), 2 (second row), and 3 (third row) in Figure 11 are shown. As it is possible to see, especially in case of targets 1 and 2, after frame 55 the number of corners that have their persistence increased (added corners and updated corners) is much larger in case of noncollaborative approach. After frame 70, when the noncollaborative approach is beginning to fail, the number of corners with increased persistence by the collaborative approach algorithm is very low and this is due to the fact that the visible part of target is very little. This is a further

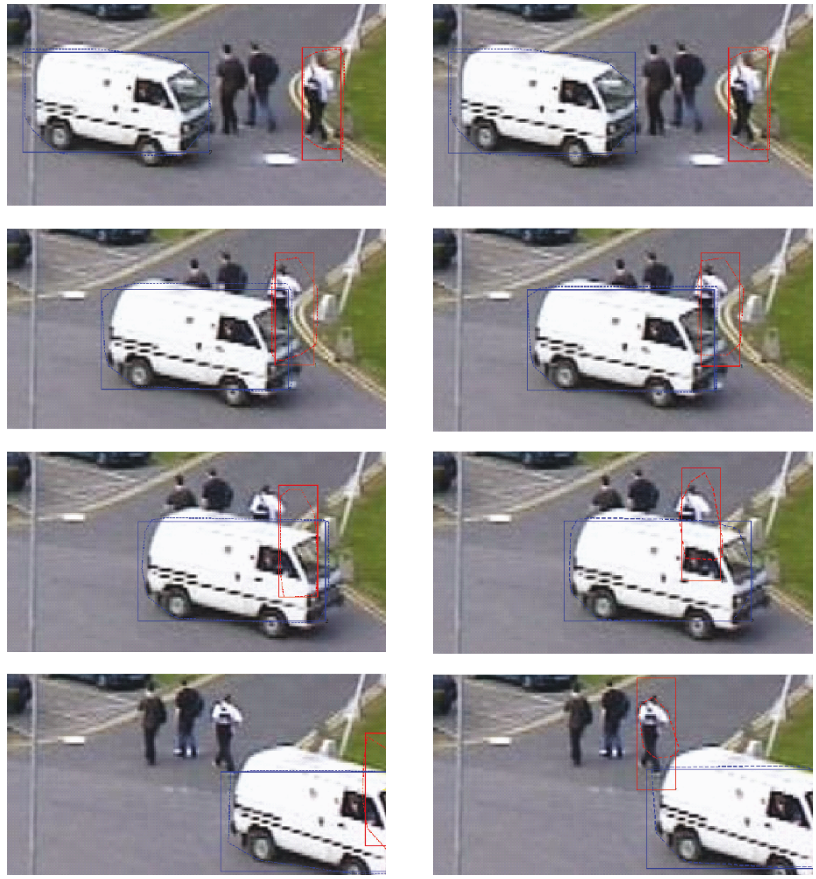


Figure 14: Sequence 3. Comparison between noncollaborative (left column) and collaborative approaches (right column).

Table 2: Tracking results.

	PETS2006 name	Number of frames	Number of objects	Short-term interaction	Long-term interaction	Failures
Sequence 4	S3-T7-A	2271	19	8	3	0
Sequence 5	S6-T3-H	2800	14	5	2	2

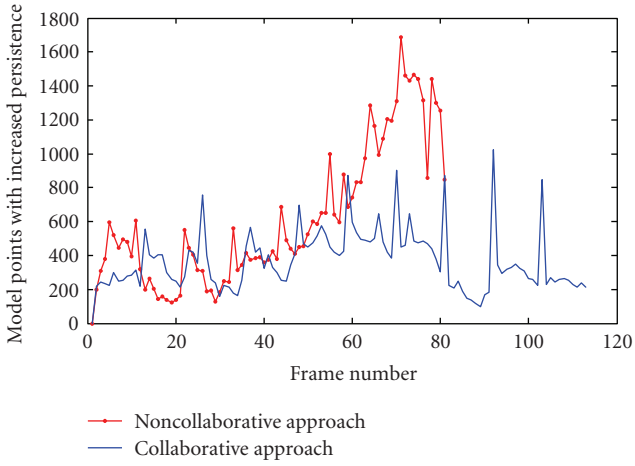


Figure 15: Sequence 2. Number of model points with increased persistence outside the manual extracted convex hull of the target.

proof of the ability of the collaborative approach with feature collaboration to adapt to the scene, even in case of heavy interaction between the tracked objects.

5.4. Sensitivity to background clutter

To evaluate the performances of the algorithm in cluttered situations, three experiments have been set up. A sequence (Sequence 6) that is correctly tracked in normal conditions has been selected (the tracking results are shown in Figure 17). The sequence has a resolution of 640×480 and is 51 frames long. In the first experiment, corners are added randomly to the list of the extracted corners simulating a random extraction from all the image plane. This experiment allows the analysis of the performance simulating the presence of random noise in the image. In case of heavy image noise, in fact, more corners are extracted by the

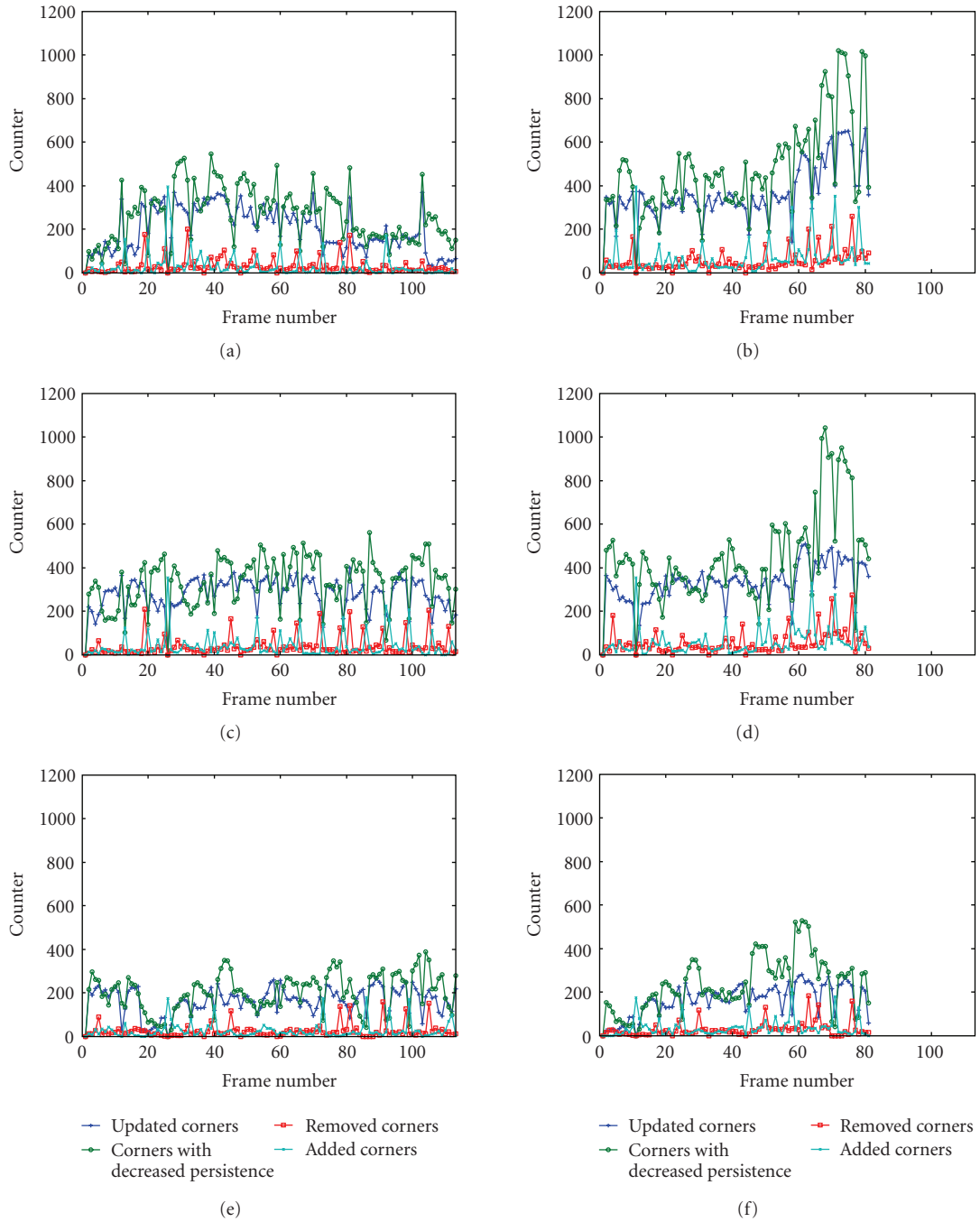
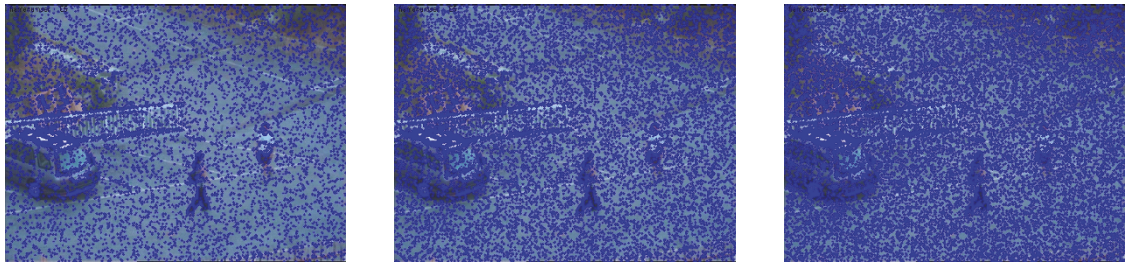


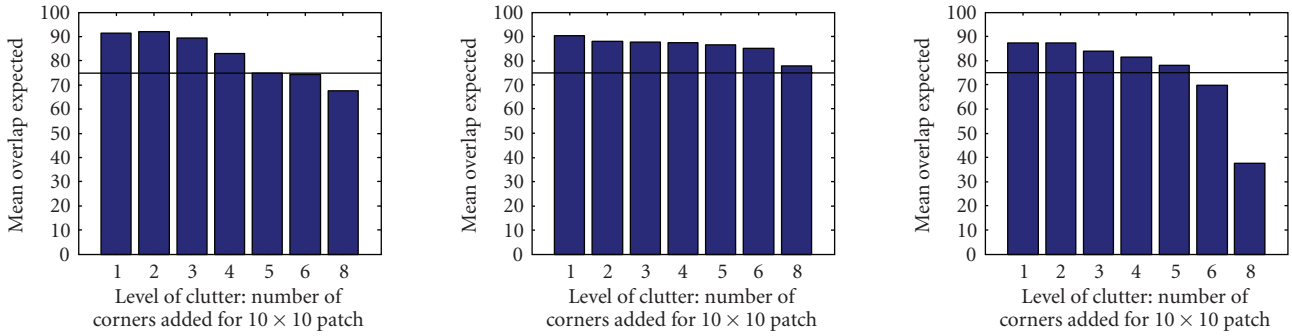
Figure 16: Sequence 2. Number of updated, decreased in persistence, removed, and added model corners in collaborative (left column) and noncollaborative (right column approaches) for targets marked as 1 (first row), 2 (second row), and 3 (third row) in Figure 11.

Figure 17: Sequence 6. Tracking results in case of no added background corners.



(a) (b) (c)

Figure 18: Sequence 6. Example of different levels of simulated clutter on Sequence 4. (a) 2 corners were randomly added for a patch of 10×10 pixels; (b) 6 corners were randomly added for a patch of 10×10 pixels; (c) 8 corners were added for a patch of 10×10 pixels.



(a) (b) (c)

Figure 19: Sequence 6. Tracking results in terms of overlap between the target and the estimated shape model for different levels of clutter in case of (a) experiment number 1; (b) experiment number 2; (c) experiment number 3.

corner extractor generating a really cluttered environment. The second experiment simulates the presence of a textured time-variant background (i.e., trees in a windy day). To do that, the corners are added randomly at each frame to the list of the extracted corners simulating the extraction from the entire image plane but the zone of the targets (that are in foreground) which was manually segmented. The third experiment finally simulates the condition of a fixed cluttered background. To simulate this condition, a random pattern of corners is generated at the beginning of the sequence. These corners are added to the list of the extracted corners, but at each frame the corners that are in the zone of the target are discarded. To evaluate the performances, the mean expected overlap between the convex hull of the estimated shape of each target and the ground truth convex hull that contains the target is estimated in different conditions of clutter by using a Monte Carlo technique. The mean number of added corners for a patch of 10×10 pixels is used to quantify the level of clutter. Some examples of the different levels of clutter are shown in Figure 18. The results for the three experiments are shown in Figure 19 where a line has been plotted to highlight the condition of 75% overlap. To better understand the reported values, it is worth saying that due to the length and symmetry of the sequence (the occlusion situation occurs in the middle), a mean overlap of 75% means that during the occlusion, the track of one target is lost. While an overlap of more than 85% means that the trackers tracked correctly and that there are no visually perceptible errors. As it is possible to see, the proposed algorithm, by exploiting the strong geometrical relations between the model points, is stable also in heavy clutter.

5.5. Comparison with other trackers

In this paragraph, our algorithm will be compared to some successful methods for multiple target tracking. In particular, the algorithm will be compared to the boosted particle filter (BPF) [11], the netted collaborative autonomous trackers (NCATs) [14], the particle based belief propagation (PBBP) [16], and the multiple hypothesis filter (MHT) [19]. Moreover, the results will be compared to the results obtained by multiple independent color-based trackers [7]. This comparison is possible thanks to the work presented in [16] and to the sequence provided by Okuma [11] on his website.

The results of the proposed method on the hockey sequence (Sequence 7) are reported in Figure 20. As it is possible to see, during all the sequence by using our approach, there are no labeling or merging errors.

To correctly compare the proposed approach with the approaches in literature, a brief description of the features used by the above-mentioned methods is here reported. Both BPF and PBBP use the simple Bhattacharyya similarity coefficient to measure the similarity between the target model

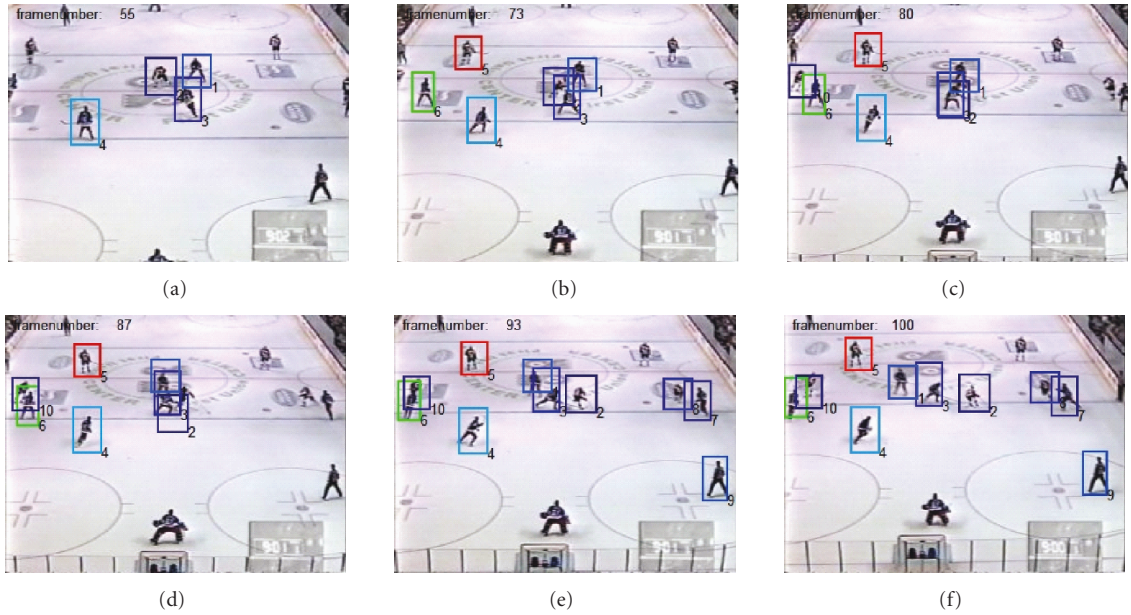


Figure 20: Sequence 7. Tracking results of the proposed method.

and the image region (likelihood computation), the prior distribution that is used to sample the particles, on the other hand, is the result of a fusion between the a priori motion model of the targets and the detections of an offline learned detector (Adaboost for BPF and SVM for PBBP). The NCAT approach uses to model the target a color-histogram trained offline in a boosted fashion. The results provided in [16] by using the MHT method are similarly obtained by using an offline-trained SVM classifier. The color-based trackers in [7] finally use a Bhattacharyya similarity coefficient and the states of the targets are propagated by using the particle filter approach.

All these methods (apart from [7]) use strong a priori information about the target to be tracked, and it is not therefore possible to track different classes of targets without retraining the classifier. The proposed approach on the other hand is general and does not need any a priori information about the target appearance or motion. It can in fact be used to track without any modification a pedestrian or a van (see Figure 14). Some quantitative results about the hockey sequence are shown in Figure 21. In this figure, the coordinates of the targets labeled as 1, 2, 3 estimated by using our approach are reported by using a solid line, the ground truth (hand labeled in [16] and here reported) is plotted by using a dashed line while the results of the PBBP approach are shown by using only the markers. While the positions of targets 1 and 3 are estimated correctly during all the sequence by the proposed method, during the complete occlusion of target 2, the position is estimated with an error of about 10 pixels for some frames but, after the occlusion, the correct position is recovered. PBBP, on the other hand, tracks the object with more precision during all the occlusion. This difference in performances is due to the fact that the proposed method uses, for each target, an independent (and in this case uniform) motion model and

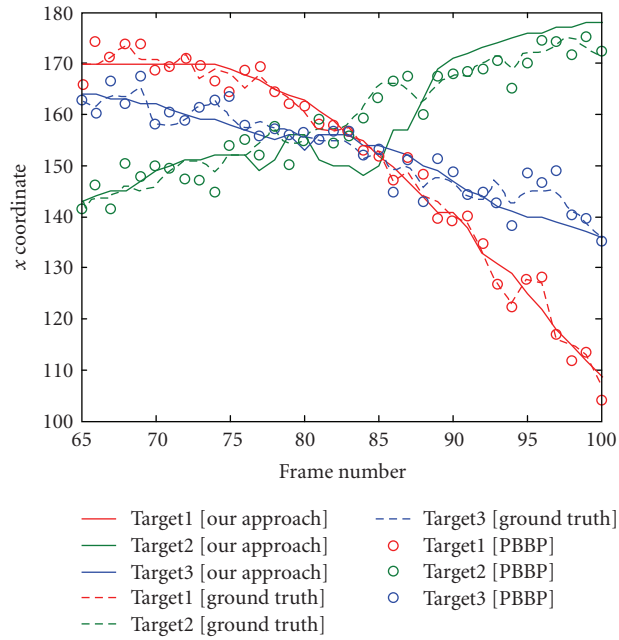


Figure 21: Sequence 7. Estimated coordinates of trackers 1, 2, and 3. Solid line: proposed approach. Dashed line: ground truth. Circles: PBBP approach [16].

the tracking results are therefore governed by observations. In case of occlusion, the proposed approach does not let the trackers use the same observation and therefore the small number of observations can cause an inaccurate estimation. On the other hand, PBBP bypasses the problem of lack of observation by modelling the motion during interaction. In [16], it is stated that among the compared trackers, PBBP obtains the best performances followed by NCAT that

obtain similar performances on the hockey sequence. BPF is less accurate than PBBP but is still capable of tracking the hockey sequence with good performances. Finally both MHT and the approach in [7] fail in tracking targets during the occlusion situation.

Summarizing our approach by obtaining results comparable to state-of-the-art methods that use offline-learned models and by outperforming methods like [7] can be the right choice when strong a priori information about the appearance or the motion of the targets to be tracked is not available.

6. CONCLUSION AND FUTURE WORKS

In this paper, an algorithm for feature classification and its exploitation for both single and collaborative tracking have been proposed. It is shown that the proposed algorithm is a solution to the Bayesian tracking problem that allows position and shape estimation even in clutter or when multiple targets interact and occlude each other. The features in the scene are classified as good, suspicious, malicious, and neutral, and this information is used for avoiding clutter or distracters in the scene and for allowing continuous model updating. In case of multiple tracked objects, when the algorithm detects an interaction between them, the collaboration and the sharing of classification information allow the segmentation and the assignment of features to the trackers. The reported experimental results showed that the use of feature classification improves the tracking results both for single- and multitarget trackings.

As a feature work, the possibility of improving the tracking performances by using an MRF approach in the shape-updating process with the aim of maintaining shape coherence and assigning observation to the correct tracker during collaboration is being evaluated. The other possibility for the future development of this method is to consider error propagation in the Bayesian filtering.

REFERENCES

[1] M. Asadi, A. Dore, A. Beoldo, and C. S. Regazzoni, "Tracking by using dynamic shape model learning in the presence of occlusion," in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '07)*, pp. 230–235, London, UK, September 2007.

[2] M. Asadi and C. S. Regazzoni, "Tracking using continuous shape model learning in the presence of occlusion," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 250780, 23 pages, 2008.

[3] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 511–519, 2007.

[4] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[5] M. Bogaert, N. Chleq, P. Cornez, C. S. Regazzoni, A. Teschioni, and M. Thonnat, "The passwords project," in *Proceedings of IEEE International Conference on Image Processing (ICIP '96)*, vol. 3, pp. 675–678, Lausanne, Switzerland, September 1996.

[6] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42–56, 2000.

[7] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*, vol. 1, pp. 661–675, Copenhagen, Denmark, May 2002.

[8] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1208–1221, 2004.

[9] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, New York, NY, USA, 1988.

[10] C. Hue, J.-P. Le Cadre, and P. Perez, "A particle filter to track multiple objects," in *Proceedings of IEEE Workshop on Multi-Object Tracking (MOT '01)*, pp. 61–68, Vancouver, Canada, July 2001.

[11] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: multitarget detection and tracking," in *Proceedings of the 8th European Conference on Computer Vision (ECCV '04)*, vol. 3021, pp. 28–39, Prague, Czech Republic, May 2004.

[12] M. Isard and J. MacCormick, "BraMBLe: a Bayesian multiple-blob tracker," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 2, pp. 34–41, Vancouver, Canada, July 2001.

[13] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different lifespans," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[14] T. Yu and Y. Wu, "Decentralized multiple target tracking using netted collaborative autonomous trackers," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 939–946, San Diego, Calif, USA, June 2005.

[15] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 789–795, Madison, Wis, USA, June 2003.

[16] J. Xue, N. Zheng, J. Geng, and X. Zhong, "Tracking multiple visual targets via particle-based belief propagation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 1, pp. 196–209, 2008.

[17] http://www.isip40.it/index.php?mod=04_Demos/sequences.html.

[18] F. Monti and C. S. Regazzoni, "Joint collaborative tracking and multitarget shape updating under occlusion situations," submitted to *IEEE Transactions on Image Processing*.

[19] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.