Real-Time Distributed Multi-object Tracking in a PTZ Camera Network

Ayesha Choudhary[1(✉)], Shubham Sharma[2], Indu Sreedevi[2], and Santanu Chaudhury[3]

[1] School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi, India
ayeshac@mail.jnu.ac.in
[2] Department of Electronics and Communication Engineering, Delhi Technological University, New Delhi, India
shubh2494@gmail.com, s.indu@rediffmail.com
[3] Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, India
santanuc@ee.iitd.ac.in

Abstract. A visual surveillance system should have the ability to view an object of interest at a certain size so that important information related to that object can be collected and analyzed as the object moves in the area observed by multiple cameras. In this paper, we propose a novel framework for real-time, distributed, multi-object tracking in a PTZ camera network with this capability. In our framework, the user is provided a tool to mark an object of interest such that the object is tracked at a certain size as it moves in the view of various cameras across space and time. The pan, tilt and zoom capabilities of the PTZ cameras are leveraged upon to ensure that the object of interest remains within the predefined size range as it is seamlessly tracked in the PTZ camera network. In our distributed system, each camera tracks the objects in its view using particle filter tracking and multi-layered belief propagation is used for seamlessly tracking objects across cameras.

Keywords: Distributed multi-camera tracking · Real-time tracking · PTZ camera network · Collaborative multi-object tracking · Belief propagation

1 Introduction

A real-time video surveillance system consisting of a PTZ (pan, tilt, zoom) camera network requires seamless tracking of multiple objects in the scene. Moreover, particular objects of interest, such as suspects, may be required to be tracked at a certain dimension in each frame so that important information related to that object is continuously retained. In general, it is possible that the object of interest can become so small that a lot of information about the object is lost. On the other hand, the object of interest can come so close to a camera that the

M. Kryszkiewicz et al. (Eds.): PReMI 2015, LNCS 9124, pp. 183–192, 2015.
DOI: 10.1007/978-3-319-19941-2_18

object becomes too large and blocks the view of the camera, hiding important information. In this paper, we propose a novel framework for real-time, distributed multi-object tracking in PTZ camera network that also addresses the situation mentioned above. The pan/tilt capability of the cameras along with camera handoff ensures seamless tracking of the objects in the scene. We leverage on the zoom capability of the cameras to ensure that the objects of interest are tracked at a certain size as the objects move across space and time in the camera network. The main contributions of our framework are: (a) multiple objects are seamlessly tracked across space and time in the camera network; (b) the user is provided with a tool to mark an object of interest, such that, the object of interest can be seamlessly tracked at a certain predefined size throughout the area under observation; (c) the user is notified when the object of interest leaves the area under observation.

Distributed PTZ camera networks are well-suited for wide area surveillance [2]. However, such a system is complex because network topology changes as cameras pan, tilt or zoom to seamlessly track the objects. We assume a distributed system with an underlying communication network such that each camera can communicate with every other camera either directly or indirectly. We assume that the camera network is calibrated and each camera has the list of its network neighbors. We define the network neighbors of a camera as those cameras that have overlapping or contiguous views in some pan/tilt/zoom position of the camera. When a camera receives a message from any of its network neighbors, it takes the decision to pan/tilt/zoom so that the object can be seamlessly tracked at the required size. Data fusion between cameras viewing a common region and across cameras needs to be addressed to enable seamless tracking. We apply belief propagation at multiple levels for data fusion.

In our framework, we assume that there are priority areas, that are prespecified. These priority areas also include the entry and exit locations in the area under observation. Placement of cameras in this case plays an important role. We apply the optimal placement algorithm [6] to place the cameras in such a manner that the priority areas are observed at all times. Since the cameras that view the entry/exit areas are static for a certain time period, these cameras apply background subtraction [12] to detect objects that enter the area under observation. Based on the detected object, we initialize the particle filter tracker [16] in these cameras. The camera then communicates the particle filter estimates of the detected object to all its network neighbors. The system ensures that as the object moves in the area under observation, it is continuously tracked at all times by at least one camera. In the next section we discuss the related work.

2 Related Work

In recent times, research on multi-camera tracking in camera networks consisting of static cameras as well as PTZ camera networks has been gaining importance [1,3]. More recently, research on active camera systems using distributed processing is gaining importance since they are better suited for wide

area surveillance [4]. Various distributed computer vision algorithms are discussed in [10,13–15]. A system consisting of static and PTZ cameras was proposed in [8] for surveillance of a parking lot. It is a hierarchical framework and uses the active camera for tracking a suspicious object at higher resolution. Authors in [4] apply distributed optimization in the game theoretic framework for controlling PTZ cameras in a wide area distributed camera network. The aim is to optimize solutions for various dynamic scene analysis problems. Moreover, the cameras collaborate among themselves to ensure that all objects are seamlessly tracked. The concept of multi-player learning in games have also been used in [11], for distributed collaboration among neighboring cameras viewing a common target for multi-object tracking in a PTZ camera network. In comparison to these systems, our framework consists of only PTZ cameras and each camera zooms in or out as required to track the pre-specified target at a certain resolution. Our framework provides a user interaction layer, to enable the user to mark objects of interest as they enter into the scene. The user can also specify the size at which the objects of interest should be tracked. Moreover, we use particle filter based tracking in each camera independently and use its parameters in multi-layered belief propagation for collaborative tracking of multiple objects in the area under observation. Authors in [9], proposed a method for controlling PTZ cameras to obtain high resolution face images of targets at opportune points in time for each camera in a distributed PTZ camera network. Our work is essentially different from this as it tracks the whole body of the targets of interest at a pre-specified size requiring the camera to zoom in or zoom out while the object is in its view. Moreover, if the camera is tracking more than one target of interest, it collaborates with the neighboring cameras to ensure that at least one camera is tracking the object at the required size.

3 Particle Filter Based Tracking Framework

Particle filter is a Monte Carlo method that is simple, yet capable of approximating complex models. Let the total number of particles be N and the total number of components be M. Then, $X_t = {x_t^{(i)}}_{i=1}^{N}$ be the particles and the particle weights are $W_t = {w_t^{(i)}}_{i=1}^{N}$. Then, the mixture filtering distribution is of the form given in [16],

$$p(x_t|z^t) = \sum_{k=1}^{M} \pi_{k,t} \sum_{i \in I_k} w_t^{(i)} \delta_{X_t^{(i)}}(x_t) \tag{1}$$

where, $\delta_b(.)$ is the Dirac delta function with mass at b and I_k is the set of indices of the particles belonging to k^{th} mixture component. These particles are updated sequentially, and the new weights are recalculated at each step. The new particle set P^t has to be computed in such a manner that it is a sample set from $p(x_t|z^t)$ given that the particle set P^{t-1} is a sample set from $p(x_t|z^{t-1})$. Each component evolves independently in the tracking module and therefore, the particle representation of each mixture component also evolves independently.

3.1 Measurement Module

In our framework, all entry locations are priority areas and therefore, continuously observed by at least one camera, that is static for a certain time period. As an object enters the area under observation, it is detected using background subtraction [12] and represented by its bounding box. The reference color model of the object is created at the time it is first detected in the manner discussed below.

Let $B_i = \{x_i, y_i, w_i, h_i\}$ denote the bounding box of the object of interest, where, (x_i, y_i) is the center of the bounding box and (w_i, h_i) denote the width and height of the bounding box respectively. Similar to [5], we consider the Hue-Saturation-Value (HSV) color histogram of the bounding box to represent the measurement model that is robust with respect to illumination changes. The HSV histogram consists of N bins where $b_t(p) \in \{1, 2, \ldots, N\}$ is the bin index at the color value $y_t(p)$ at pixel location p in frame t. The HSV histogram is formulated for the pixels inside the bounding box and the kernel density estimate is $H(x_t) \triangleq \{h(n; x_t)\}$, $n = 1, 2, \ldots, N$ of the color distribution at time t is given by

$$h(n; x_t) = \alpha \sum_{d \in B_i} \delta[b_t(p) - n] \tag{2}$$

For tracking, in each frame the color model of the previous frame is treated as the reference color model, H^*, to overcome the variations in the background as the objects moves in the scene. Similar to [5], the distance between the reference color model and the color model of the current frame is calculated using the Bhattacharya distance given by Eq. 3:

$$d(H^*, H(x_t)) = \left[1 - \sum_{n=1}^{N} \sqrt{h * (n; x(t-1))h(n; x_t)}\right]^{\frac{1}{2}} \tag{3}$$

The likelihood distribution that is required for particle filter tracking is obtained using the distance between the current and previous HSV color histograms given by Eq. 4:

$$p(z_t|x_t) = \gamma e^{-\beta d^2(H^*, H(x_t))} \tag{4}$$

where, γ and β are normalizing constants.

3.2 Single Camera Tracking

In our distributed framework, each camera tracks the objects in its view based on its own image measurements. These measurements are measured as described in Sect. 3.1 and used to initialize the particle filter tracker. Let $x_{o:t} = \{x_0, x_1, \ldots, x_t\}$ be the state vector and $z_{0:t}$ be the observation vectors up to time t. Then, for a particular object O_i, the posterior probability distribution is given by Eq. 5

$$p_i(x_t|z_{0:t}) = \frac{p(z_t|x_t)p(x_t|z_{0:t-1}))}{p(z_t|z_{0:t-1})}$$
$$= p(z_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})dx_{t-1} \quad (5)$$

Given that there are M objects in a camera's view, then the posterior distribution $p(x_t|z_{0:t})$ is modeled as an M-component non-parametric mixture model given by Eq. 6

$$p(x_t|z_{0:t}) = \sum_{i=1}^{M} \pi_{i,t} p_i(x_t|z_{0:t}) \quad (6)$$

where, the weights $\pi_{i,t}$ are such that $\sum_{j=1}^{M} \pi_{j,t} = 1 \forall t$. As can be easily seen from Eq. 5,

$$\pi_{i,t} = \frac{\pi_{i,t-1} \int p_i(z_t|x_t)p_i(x_t|z_{0:t-1})dx_t}{\sum_{j=1}^{M} \pi_{j,t-1} \int p_j(z_t|x_t)p_j(x_t|z_{0:t-1})dx_t} \quad (7)$$

There are M different likelihood distributions $p_k(x_t|z_{0:t}), k = 1, 2, \ldots, M$, one for each object in the cameras view. Usually, in multi-camera tracking, it is assumed that all the M objects are being viewed by all the cameras, however, in our framework, this does not hold true. Since the camera network is spread in the area under observation, it is not necessary that even two cameras will be viewing the same area. Therefore, each camera computes these likelihood distributions for the objects that are in that camera's view. The 3D position of the object gives the identity of the object since that is a unique feature for each object. Only one object can be in a 3D position at a time. Moreover, each camera will have its own uncertainty in measurement, so the 3D position is taken to be same if it is within a predefined threshold.

4 Collaboration for Multi-camera Tracking

In this section, we assume that each camera is capable of tracking multiple objects in its view. The same object may or may not be tracked by more than one camera simultaneously. However, since the camera network is calibrated, the 3D position of the object can always be calculated. Since each object is represented by its bounding box and its center (x, y), each camera computes whether an object will get out of its view or not. In general, a camera pans and/or tilts to keep the object in its view, however, there are limitations on the maximum pan/tilt that a camera can perform. Therefore, when an object is about to get out of the view of the camera, that camera sends a message to all its neighboring cameras about this object. The message contains all the information, such as, current 3D position, size of the bounding box, the track till the current point in time, the probability estimates till that time as well as the predicted 3D position of that object.

A camera is a neighbor of another camera if an object can get out of one camera's view and get into the other camera's view, or if both the cameras

have overlapping views in some pan/tilt position of both the cameras. When a camera receives a message from its neighbor, it checks whether it is tracking the same object or not. If it is tracking the same object, it continues to do so. If the message is about a new object, the camera checks whether the object has entered its view or not. To check the identity of the object, we use the 3D location of the center of the bounding box. The camera first checks whether it is already tracking that object. It computes the distance between the $3D$ position received in the message with the $3D$ position of the objects that it is currently tracking. If this distance is within a threshold with one of the objects in the camera's view, it identifies that the message received is for an object that is currently being tracked and continues to track that object. In case, the distance is not within the threshold, it checks whether a new object has entered its view. To do so, the camera that receives the message, computes the image coordinates of the 3D position of the object, and considers a bounding box around that image position. It then forms an HSV color histogram of the pixels in that bounding box and compares the distance between the histogram received in the message with the computed histogram. If the distance is within a predefined threshold, it assumes that the object has been identified. It continues to track the object using the information present in the message.

Belief propagation is used to compute the probabilities in the new view, based on the probabilities received from the camera that was previously tracking it. Let C_k be the camera that has received messages about an object from multiple cameras, $j = 1, 2, \ldots, r$, where r could be 1 or more than 1. Then, each C_j in its message also sends the predicted value of the object, that is, $x_{t,j}$, $j = 1, 2, \ldots, r$. Let the target state in C_k be $x_{t,k}$ and its state in each C_j be $x_{t,j}$, $j = 1, 2, \ldots, r$.

Let $z_{t,j}, j = 1, 2, \ldots, r$ denote the observation in C_j at time t. Then, $Z_t = \{z_{t,1}, \ldots, z_{t,r}\}$ be the multi-camera observation at time t. This implies that $Z^t = \{Z^1, Z^2, \ldots, Z^r\}$ are the multi-camera observations till time t. Then, the message from camera C_j to C_k is

$$m_{kj}(x_{t,k}) \leftarrow p_j(z_{t,j}|x_{t,j})\psi^t_{k,j}(x_{t,k}, x_{t,j}) \times \int p(x_{t,j}|x_{t-1,j})p(x_{t-1,j}|Z^{t-1})dx_{t-1,j}dx_{t,j} \tag{8}$$

Then, the belief is computed by,

$$p(x_{t,k}|Z^t) \propto \Pi_{j=1,\ldots,r} m_{kj}(x_{t,k}) \times \int p(x_{t,k}|x_{t-1,k})p(x_{t-1,k}|Z^{t-1})dx_{t-1,k} \tag{9}$$

where, $p(x_{t,k}|x_{t-1,k})$ is computed as discussed above and $p(x_{t-1,k}|Z^{t-1})$ is set to 1, since the object was not in this camera during that time period.

Therefore, even if the camera has not seen the object of interest before it will get tracked using the history from its neighboring cameras.

5 Zooming into an Object

In our framework, we give the user the capability to observe any person such that the size of the object in any camera's view is within a predefined range, as he/she moves across the camera network. The user can mark the object of interest when he/she enters the area under observation. Then, as the object moves across the camera network, along with the message that each camera sends to its neighbors, a tag is also sent and the range of the size of the tracked object.

This ensures that each camera can change its pan/tilt and zoom parameters to continuously track the object such that the size of that object in that camera's view is within the predefined range. It is not necessary that the object remains in the center of the image, therefore, the camera does not need to pan/tilt continuously. Instead, the camera needs to change its pan/tilt and zoom to be able to track the object at the required size. By size of the object, we imply the size of the bounding box. In many cases, if the camera zooms without bringing the object to its center, then it may lose the object from its zoomed view. Therefore, before zooming, the camera pans and/or tilts to bring the object to the center of its view and then zooms into be able to view the person at the predefined size. The camera only pans and/or tilts next when the object is about to get out of its view or if the size of the object goes out of the desired range.

Suppose that object O_i is currently in view of camera C_j and about to get in the view of its neighbor, C_k. Then, C_k also receives the predicted $3D$ position of O_i. Once C_k checks that O_i is within its view, it checks on the size of the object. If the size is outside the required range, then first the camera C_k pans/tilts to bring the object into its center and then zooms. This is to ensure that the object is not lost from the camera's view after zooming. Panning by angle α is rotation about the Y-axis by α and tilting by angle β is rotation around the X-axis by β. Let (X_i, Y_i, Z_i) be the position of O_i in C_k. Then, as discussed in [7],

$$\alpha = -\arctan \frac{X_i}{Z_i} \tag{10}$$

and,

$$\beta = \arctan \frac{Y_i}{Z_i \cos\alpha - X_i \sin\alpha} \tag{11}$$

After the target object is centered, the camera zooms in by $\delta = f'_k - f_k$ where, f'_k is the focal length of C_k after zoom in.

Then, the zoom f'_k is computed as

$$f'_k = \frac{aH_i}{H} Z_i \tag{12}$$

where, a is the ratio of the current height h_i to the desired height H_i, $\frac{h_i}{H_i} \leq a \leq 1$ and H is taken to be the average human height. And, tracking is resumed after adjusting the size of the target. Since this does not take too much time, the tracking is smooth despite the transition.

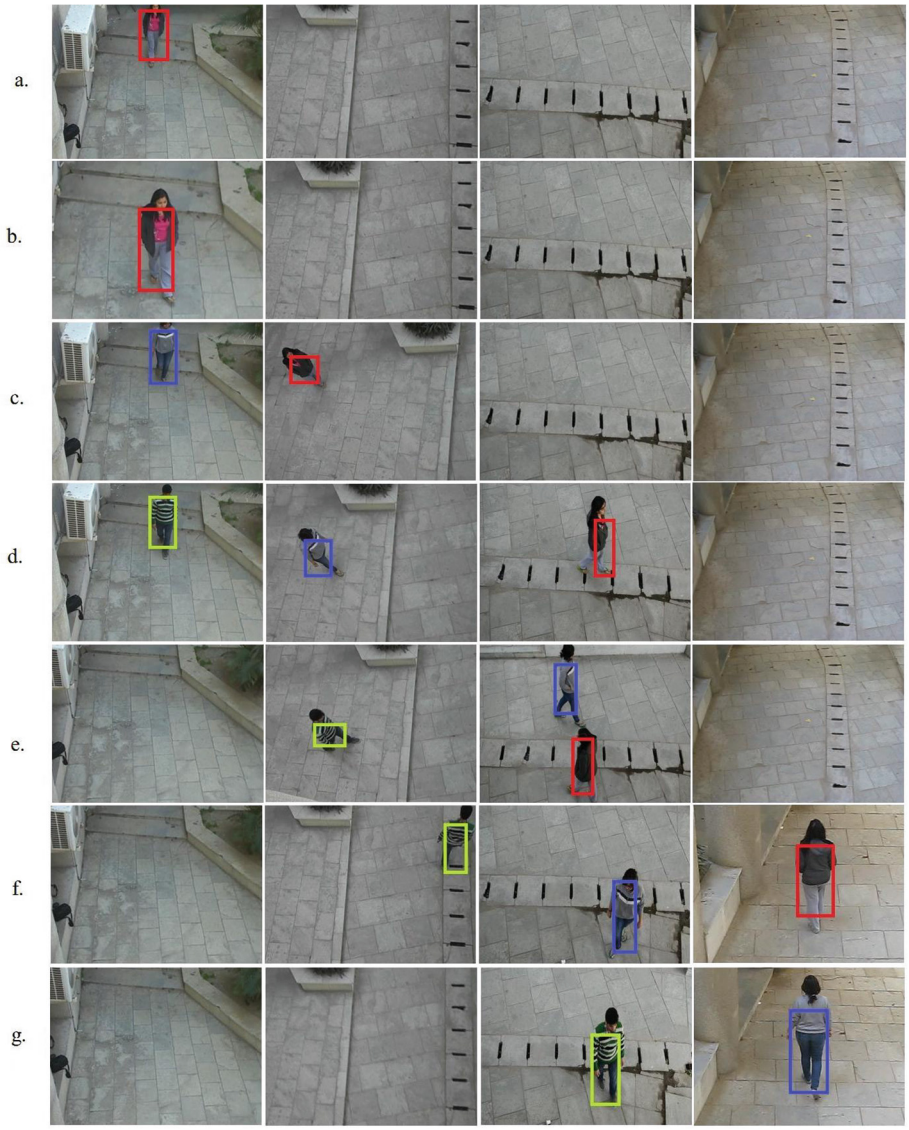

Fig. 1. Each row shows the state of the four cameras C1, C2, C3, C4, respectively at different time stamps. Two objects of interest O1 (red) and O2 (blue) are in the scene. In(a), O1 enters at C1 and the user marks it as an object of interest. (b) C1 zooms to bring O1 to the predefined size. (c) O2 enters, since C1 is about to zoom, it will lose O1. C1 communicates to its neighbor C2 and C2 pans to bring O1 in its view. (d) O3 enters and O1 and O2 are tracked by C3 and C2 respectively. (e) O1 and O2 are in same cameras but moving in different directions. Therefore, in (f) C4 pans and zooms to track O1, while C3 continues to track O2. (g) O1 has exited the scene, that is informed to the user, O2 is also an object of interest and therefore, tracked at the zoom level (Color figure online).

6 Experimental Results

We perform various experiments using four PTZ SONY EVI D70 cameras, $C1$, $C2$, $C3$ and $C4$. In the scene, camera $C1$ views the entrance and camera $C2$ views the exit. These are the two priority areas. We show one of the scenarios of our experimentation that covers the all aspects of our framework. In Fig. 1, the user marks two objects of interest $O1$ (red) and $O2$ (blue) when they enter the view of camera $C1$. In both cases, the camera zooms to track the objects at the predefined size. In Fig. 1(e), both the objects of interest are in the view of the same camera but moving in different directions. Since both $O1$ and $O2$ need to be tracked at all times, $C3$ sends a message about $O1$ to camera $C4$ and then, $C4$ pans, tilts and zooms to continue tracking the $O1$ at the required size.

7 Conclusion

In this paper, we have proposed a novel framework for real-time, distributed, multi-object tracking in a PTZ camera network. In our framework, the user is given the ability to mark an object of interest to track it across cameras such that the size of the object remains within a pre-specified range. If the size of the object reduces or increases beyond this range, the camera zooms in or out, as required, to bring the object's size within the range. We have used particle filter based tracking for tracking objects in each camera and multi-layered belief propagation for seamlessly tracking objects across cameras. The pan, tilt and zoom capabilities of each camera are used whenever required for seamlessly tracking all the objects in the scene.

References

1. Black, J., Ellis, T.: Multi-camera image tracking. Image Vis. Comput. **24**(11), 1256–1267 (2006)
2. Choudhary, A., Sharma, G., Chaudhury, S., Banerjee, S.: Distributed calirbration of a pan-tilt camera network using multi-layered belief proapagation. In: Proceedings of IEEE Workshop on Camera Networks in conjunction with CVPR (2010)
3. Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multisensor surveillance. Proc. IEEE **89**(10), 1456–1477 (2001)
4. Ding, C., Song, B., Morye, A., Farrell, J., Roy-Chowdhury, A.: Collaborative sensing in a distributed PTZ camera network. IEEE Trans. Image Process. **21**(7), 3282–3295 (2012)
5. Hue, P.P.C., Gangnet, J.V.M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
6. Indu, S., Chaudhary, S., Mittal, N., Bhattacharya, A.: Optimal sensor placement for surveillance of large spaces. In: Indian Conference on Vision, Graphics and Image Processing (ICVGIP) (2008)
7. Lu, Y., Payandeh, S.: Cooperative hybrid multi-camera tracking for people surveillance. Can. J. Electr. Comput. Eng. **33**(3/4), 145–152 (2008)

8. Micheloni, C., Foresti, G.L., Snidaro, L.: A network of cooperative cameras for visual surveillance. Vis. Image Signal Process. **15**(2), 205–212 (2005)
9. Morye, A., Ding, C., Roy-Chowdhury, A., Farrell, J.: Distributed constrained optimization for bayesian opportunistic visual sensing. IEEE Trans. Control Syst. Technol. **22**(6), 2302–2318 (2014)
10. Song, B., Ding, C., Kamal, A.T., Farrell, J.A., Roy-Chowdhury, A.K.: Distributed camera networks. Signal Process. Mag. **28**(3), 20–31 (2011)
11. Soto, C., Song, B., Roy-Chowdhury, A.K.: Distributed multi-target tracking in a self-configuring camera network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
12. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE International Conference on Computer Vision and Image Proccessing (1999)
13. Taj, M., Cavallaro, A.: Distributed and decentralized multicamera tracking. Signal Process. Mag. **28**(3), 46–58 (2011)
14. Tron, R., Vidal, R.: Distributed computer vision algorithms. Signal Process. Mag. **28**(3), 32–45 (2011)
15. Tron, R., Vidal, R.: Distributed computer vision algorithms through distributed averaging. In: Proceedings of IEEE Conference on CVPR, pp. 57–63 (2011)
16. Vermaak, J., Perez, A.D.P.: Maintaining multi-modality through mixture tracking. In: International Conference on Computer Vision (2003)