Chapter 10
Multi-view Multi-object Detection and Tracking

Murtaza Taj and Andrea Cavallaro

Abstract. Multi-view trackers combine data from different camera views to estimate the temporal evolution of objects across a monitored area. Data to be combined can be represented by object features (such as position, color and silhouette) or by object trajectories in each view. In this Chapter, we classify and survey state-of-the art multi-view tracking algorithms and discuss their applications and algorithmic limitations. Moreover, we present a multi-view track-before-detect approach that consistently detects and recognizes multiple simultaneous objects in a common view, based on motion models. This approach estimates the temporal evolution of objects from noisy data, given their motion model, without an explicit object detection stage.

10.1 Introduction

Object detection and tracking is a fundamental task in various video-based applications such as security, sport analysis and tele-collaboration. Because occlusions and limited field of view make detection and tracking a challenging task, multiple video cameras can be used to increase the observability of objects, thus facilitating their consistent identification over time. Multi-camera tracking aims to establish the spatio-temporal correspondence of the same object across multiple views.

The modeling of the multi-view tracking problem depends on the management policy, the network type and the coverage of the network. The management policy of a network can be centralized [1], distributed [2] or hybrid [3]. The network can be composed of passive cameras [1], active cameras [4], or a combination of both.

Murtaza Taj
Queen Mary University of London, UK
e-mail: murtaza@elec.qmul.ac.uk

Andrea Cavallaro
Queen Mary University of London, UK
e-mail: andrea.cavallaro@elec.qmul.ac.uk

R. Cipolla, S. Battiato, & G.M. Farinella (Eds.): Computer Vision, SCI 285, pp. 263–280.
springerlink.com

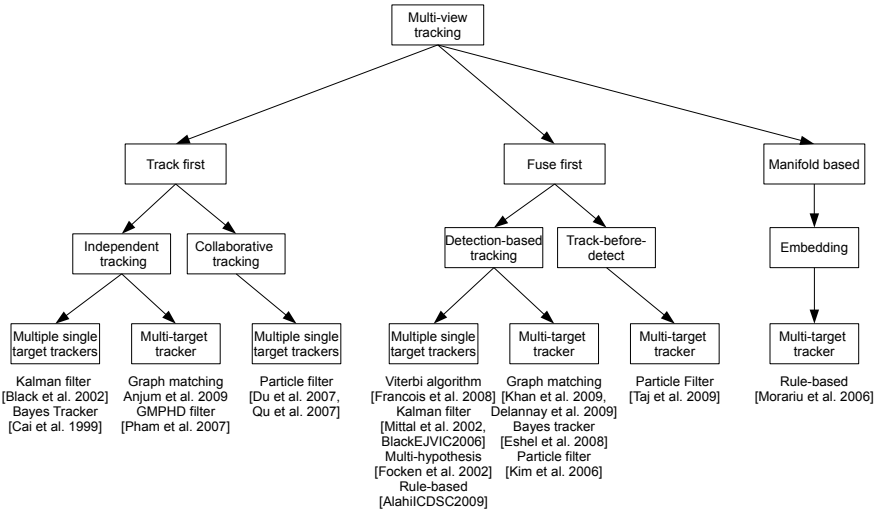


Fig. 10.1 Overview of multi-view tracking approaches.

As for the coverage of the network, the cameras can have partially overlapping [5] (multi-view) or non-overlapping [6] fields of view. This Chapter will focus mainly on tracking algorithms for partially-overlapping passive camera networks working with a centralized management policy, although some algorithms could be extended to work in a distributed fashion or with active cameras.

Algorithms for target tracking in multi-view camera networks can be grouped based on the modalities for tracking and information fusion and can be categorized into three main classes, namely *track-first*, *fuse-first* and *manifold-based*. The categorical overview of these approaches is shown in Figure 10.1. Track-first approaches perform tracking in each camera view and then project and link the resulting information on other views. Fuse-first approaches project detection information from each view to a common view and then apply tracking. Track-first approaches are in general more complex computationally but require a lower data transfer load. Track-first and fuse-first classes will be discussed in details in the rest of the Chapter.

Manifold-based approaches can be used when camera calibration information is not available, cannot be computed efficiently, or the assumption that the world is planar is not applicable. In this category, multi-camera tracking can be performed by projecting features on a manifold through Locally Linear Embedding [7]. The approach uses Caratheodory-Fejer (CF) interpolation theory, which is robust against model uncertainty and occlusion, to identify the dynamic evolution of the data on the manifolds. This method assumes that multiple views are highly overlapping and uses rule-based multi-target tracking with multiple hypotheses. The approach relies heavily on the training that uses segmented foreground objects.

The Chapter is organized as follows. The problem of multi-camera tracking is formulated in Section 10.2. Section 10.3 discusses the calibration and data fusion

between multiple views using plane-to-plane homography. Track-first approaches are discussed in Section 10.4 that covers methods using independent trackers and collaborative trackers. Fuse-first approaches are described in Section 10.5 that covers detection-based tracking as well as simultaneous detection and tracking. Finally, in Section 10.6 we draw the conclusions.

10.2 Problem Formulation

Let a wide area be monitored by a set $C = \{C_1, \cdots, C_c, \cdots, C_N\}$ of N cameras. Let $\mathbf{x}_k^{c,i}$ be the state of the i^{th} object in camera C_c and let $\mathbf{x}_k^{\pi,i}$ be the state of the i^{th} object on the common view plane π.

The state $\mathbf{x}_k^{c,i}$ can be defined based on a set of features, such as the position and the velocity components of the target in the image plane, the width and the height of the bounding box (or the axes of the ellipse) defining the area of the target, and a representation of the appearance (such as the color histogram) of the target [8].

As mentioned in Section 10.1, the multi-camera tracking problem can be categorized into two classes, namely track-first or fuse-first.

- *Track-first* approaches can be divided into four steps:
  1. Target *localization* in each view. This step extracts the localization information or measurement $Z_k^c = \{z_m^c | m = 1, \cdots, k\}$ in each view.
  2. Target *state estimation*, $\mathbf{x}_k^{c,i}$, in each view, given the set of measurements Z_k^c up to time k and the state $\mathbf{x}_{k-1}^{c,i}$ at previous time $k-1$.
  3. State estimates *projection* to a common view (from individual views). This step projects the tracks from the image views to a common view π, using the projection matrix $\mathbf{H}^{c,\pi}$, which performs a mapping from camera C_c to π:

$$\mathbf{x}_{1:k}^{\pi,i} = \mathbf{H}^{c,\pi} \mathbf{x}_{1:k}^{c,i}, \tag{10.1}$$

     where $\mathbf{x}_{1:k}^{\pi,i}$ is the projection of track $\mathbf{x}_{1:k}^{c,i}$ from camera C_c. Note that π can be the camera view selected as reference view [9, 10] or a hypothetical top view [11, 12, 13, 14].
  4. *Correspondence* resolution between projections from multiple views. This step establishes the link between all the tracks $\mathbf{x}_{1:k}^{\pi,i}$, projected from different views, belonging to the same object. The fused tracks can be reprojected to the individual views for improving track estimates.

- *Fuse-first* approaches can be divided into three steps:
  1. Target *localization* in each view. This step extracts the localization information or measurement $Z_k^c = \{z_m^c | m = 1, \cdots, k\}$ in each view.
  2. *Projection* of localization features from each view to a common view. This step projects the localization information or measurement to π:

$$Z_k^{\pi,c}(u,v) = \mathbf{H}^{c,\pi} Z_k^c(x,y); \tag{10.2}$$

and fuses them:

$$Z_k^{\pi}(u,v) = \zeta(\{Z_k^{\pi,c}(x,y)\}_{c=\{1,\cdots,N\}}), \tag{10.3}$$

where ζ is a function that fuses the measurements from multiple cameras.

3. *State estimation* in the common view. This step estimates the state $\mathbf{x}_k^{\pi,i}$ of each object in Z_k^{π}. The state can be estimated using traditional detection and tracking schemes or via simultaneous detection and tracking.

10.3 Calibration and Fusion

To fuse the data from multiple views, track-first and fuse-first approaches assume the availability of camera calibration information. Homographic transformation matrices are generally used to this end. Homographies can be computed manually [15] or automatically [16] by identifying corresponding points between views.

The automatic selection of corresponding points (auto-calibration) may be obtained through trajectory correspondence, field-of-view lines or feature-point correspondence. *Trajectory correspondence* can be achieved with a least mean square search on trajectory points from multiple views [10] or by using features such as position, velocity, size and color [17]. Automatically recovered *field-of-view lines* use the correspondence between objects in multiple views when they enter or exit the scene (i.e., when they appear on field-of-view lines in overlapping views). Both trajectory-based and field-of-view-lines-based approaches rely heavily on detection and tracking performance and assume that reliable tracks are available from each camera. Auto-calibration can also be performed using *feature-point correspondence*, for example using SIFT features followed by RANSAC to reject outliers [16]. The limitation of this approach is the assumption that the ground plane in each view is sufficiently textured in order to facilitate a reliable point correspondence.

The calibration information can then be used to map information from one view to another, using single or multi-level homography.

10.3.1 Single-Level Homography

Different features, such as points or segmentation masks, can be projected on the common view. In case of *point* projection (e.g., feet location [5, 18] or blob centroid [19]) a binary signal identifies the points (Figure 10.2), thus making this approach very sensitive to detection errors in a view. Although the error can be reduced with a Gaussian Kernel on the common view [20], these approaches are not applicable in crowded scenes, as feet or centroid locations may not be visible or may be misleading due to occlusions. In crowded scenarios a preferred solution is to track head locations that can be obtained by projecting the whole information represented by the change *segmentation mask*. In this case, $Z_k^{\pi}(u,v)$ can be obtained by computing the variance at each pixel [21] as

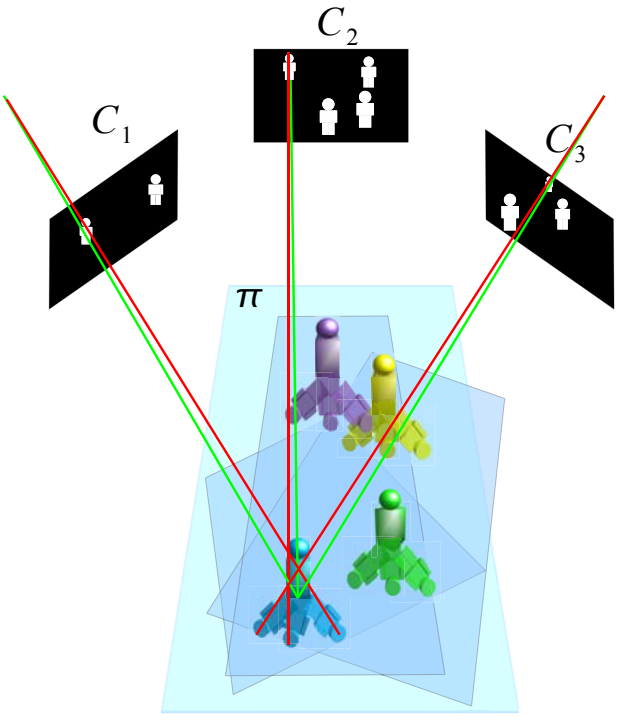


Fig. 10.2 Projection of the detections from multiple views to the top view.

$$Z_k^{\pi}(u,v) = \frac{1}{\sigma^2(\{Z_k^{\pi,c}(u,v)\}_{c=\{1,\cdots,N\}})}, \tag{10.4}$$

where

$$Z_k^{\pi,c}(u,v) = \begin{cases} \mathbf{H}^{c,\pi} Z_k^c(x,y) & if \bar{Z}_k^c(x,y) = 1 \\ 0 & otherwise \end{cases}. \tag{10.5}$$

$\bar{Z}_k^c(x,y)$ is the foreground binary mask value at (x,y) in C_c that is projected to (u,v) in $Z_k^{\pi,c}$, a single channel image.

Similarly, instead of the actual pixel values [22, 23], the binary mask values can be projected:

$$Z_k^{\pi}(u,v) = \sum_{c=1}^{N} Z_k^{\pi,c}(u,v), \tag{10.6}$$

where

$$Z_k^{\pi,c}(u,v) = \begin{cases} 1 & if \bar{Z}_k^c(x,y) = 1 \\ 0 & otherwise \end{cases}. \tag{10.7}$$

As the aforementioned approaches use a foreground binary mask, they perform prior thresholding on the image plane and may therefore ignore low-contrasted or small targets.

An alternative approach is to project the *motion estimation likelihood* without any thresholding. In this case, one can compute on the common view the product of the likelihood values from each camera [16]:

$$Z_k^{\pi}(u,v) = \prod_{c=1}^{N} Z_k^{\pi,c}(u,v), \tag{10.8}$$

where $Z_k^{\pi,c}(u,v)$ is the projected likelihood value. The drawback of this approach is that, instead of just the foreground pixels, the entire likelihood image from each view has to be projected for each time k on the common view, thus increasing the computational load.

Finally, as target points or features from more than one view can be projected on the same pixel position on the common view, each point (u,v) in $Z_k^{\pi}(u,v)$ has to be normalized with respect to the number of overlapping cameras in that region.

10.3.2 Multi-Level Homography

To increase the amount of discriminative information in the projection, one can compute the homography from multiple planes that are parallel to the ground plane [24]. Such homographies can be obtained by moving along the vertical vanishing points and then estimating projection planes that are parallel to the planar top view [16].

Let $\mathbf{H}^{c_j\pi_j}$ be the homographic matrix that projects points from c_j, the j^{th} plane in the camera C_c, to the j^{th} common-view plane π_j as

$$Z_k^{\pi_j}(u,v) = \mathbf{H}^{c_j\pi_j} Z_k^{c_j}(x,y). \tag{10.9}$$

The projections on multiple planes can either be treated separately to obtain the information about the object shape [16] or can be combined as mentioned for the single-level homography by concatenating the feature vectors from each parallel plane [22]. Figure 10.3 shows an illustration of the projection of the localization information from a camera view to multiple planes on the common view. The common view can be generated through the fusion of the pixel values from three homography planes, one at the feet level, one at the head level and one between these two planes. The fusion of the pixel values can be performed using Equation 10.4 that creates a variance map.

The signal intensity at each position is proportional to the number of foreground pixels being projected onto that position. In a multi-level homography, pixels representing different portions of an object (e.g., a person) in the image view along the vertical-axis (e.g., feet, legs, torso, neck and head) are projected around the same position on the common view, thus increasing the signal intensity.

The signal strength depends upon the number of cameras observing that region, as points contributed from multiple cameras are projected on the same location on

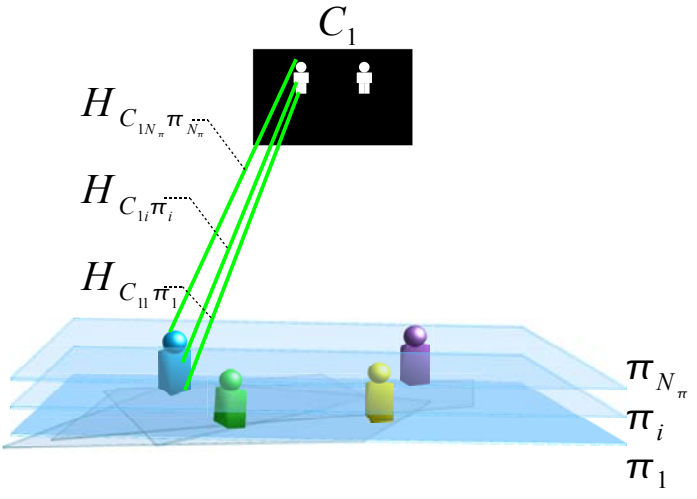
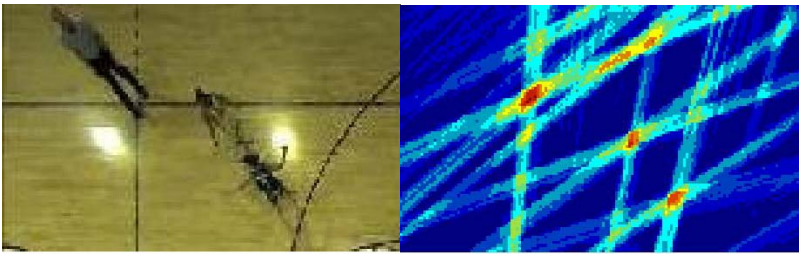


Fig. 10.3 Detections projected from one view to multiple parallel planes.



(a) (b)

Fig. 10.4 Example of parallax error. (a) Top view with 3 targets. (b) 3 high-intensity regions on the top view generated by the projections of the targets together with several other high-intensity regions due to *phantoms*.

the common view. This compensates for possible miss-detections in some camera views. However, when an object is projected on the plane, pixels not belonging to that plane are also projected there, thus creating a *shadow* of the object along that plane. These projected shadows (from multiple objects) can overlap with each other and create false signal intensities. These noise components, referred to as *parallax errors* [22] or *phantoms* [25] (Figure 10.4), have to be filtered by the tracking algorithm, as discussed in the next sections.

10.4 Track-First Approaches

Track-first multi-view tracking can be performed either *independently* in each view or *collaboratively* across views. In collaborative tracking, estimated tracks in the

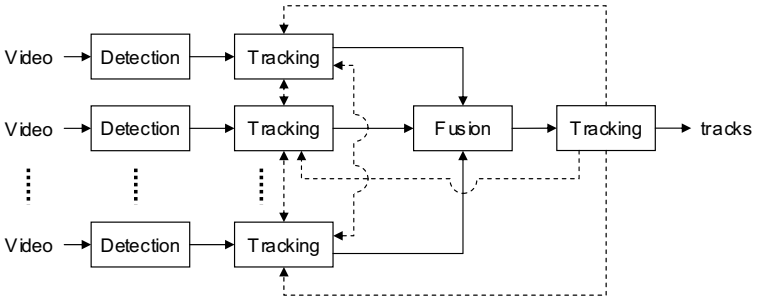


Fig. 10.5 Generic block diagram of track-first multi-view tracking algorithms. Solid line: independent tracking. Solid and dotted line: collaborative tracking.

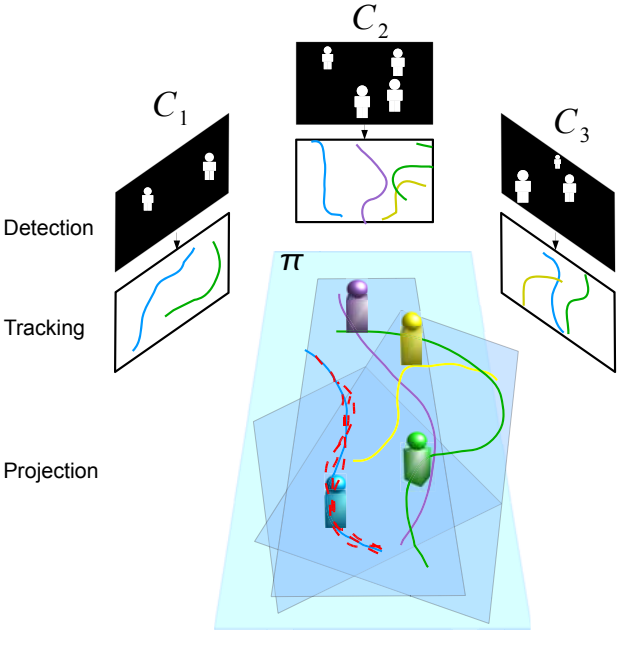


Fig. 10.6 Illustration of the track-first approach.

image view and in the common view can be used to assist each other and to improve track estimates in one view (Figure 10.5). Both independent and collaborative algorithms first track objects in each camera view and then project the tracks onto the common view for fusion (Figure 10.6). The problem to be solved here is the fusion of the multiple tracks belonging to the same target.

10.4.1 Independent Tracking

Independent tracking computes the projection of single-view tracks to another camera view [10] or to the hypothetical top view [11].

Kalman filter state estimates on the image plane can be used for single-target tracking on the top view using a second Kalman filter based on covariance mapping [10]. For multi-target tracking, independent tracks from each view are projected on the common view or on the top view for fusion. The challenge is that multiple corresponding tracks may not overlap with each other in time and space. In fact, targets may be visible in one camera during a certain time interval and in another camera during another interval. This problem can be solved by trajectory association using multiple spatio-temporal features with an off-line processing that allows recovering from failures due to occlusions and target merging [11].

A Gaussian Mixture PHD filter (GMPHD) can be used for on-line multi-target tracking using independent trackers [13]. GMPHD can be applied on each view as well as on the top view for track estimation, using features such as position, size and color histograms. The 2D estimates of the target state from each view can be projected onto the top view and used as observations for the GMPHD filter. Tracking can be performed by assigning a label to each Gaussian component. Approaches based on the PHD filter are computationally efficient as the complexity increases only linearly with the number of targets.

As estimates in a view can be affected by partial occlusions, the drawback of independent tracking is that the tracking in one view does not help improving tracking results in another view. An alternative solution is to perform collaborative tracking by using track estimates from a view as measurement for other views, as discussed in the next section.

10.4.2 Collaborative Tracking

In collaborative tracking, a set of measurements from a view are used to improve tracking results in other views.

Objects can first be tracked using a particle filter in each view and then the particles can be projected onto the top view for fusion [12]. To compute the precise location of the target on the top view, the principle axis[1] of the target can be defined in each view and then projected on the top view. The intersection of the projected principle axes can be used as the target location. The closeness of the particle to the principle axis is used as the likelihood criterion in the particle filter. To improve the results on individual views using top-view tracking, the particles in each view can be sampled from both camera-view particles and top-view particles [12].

Similarly, multiple independent regular particle filters (MIPFs) can be used to track each target in a view. The posterior in each camera can be computed by using

[1] The principle axis is the vertical line from the bottom (e.g., the feet of a person) to the top (e.g., the head of a person) of a target.

Table 10.1 Track-first multi-camera tracking algorithms. (Key: GMPHD = Gaussian Mixture Probability Hypothesis Density; MT = Multi-target tracker; IT = Independent tracking; CT = Collaborative tracking; M = Manual)

	Ref.	Features	Tracker	Calib.	MT
IT	[10]	2D position, size, velocity	Kalman filter	M	No
	[9]	2D position, height and intensity	Bayes tracker	M	No
	[11]	2D position, size, velocity	Graph matching	M	Yes
	[13]	position, size and color histogram	GMPHD filter	M	Yes
	[26]	2D position	Template matching	M	Yes
CT	[12]	2D position, size	Particle filter	M	No
	[14]	5D state space using ellipses	Particle filter	M	No

the measurements from all the cameras [14]. A summary of state-of-the-art track-first approaches is shown in Table 10.1.

Track-first approaches involve multiple tracking steps and hence can be computationally expensive. To reduce the complexity, fuse-first approaches can be used that defer the tracking step until when the information from each view is fused on a common view.

10.5 Fuse-First Approaches

Although collaborative track-first approaches help improving trajectory estimation in each camera view, they involve multiple tracking steps that can introduce sources of estimation error. These multiple steps can be eliminated by tracking on the common view only, by accumulating on the common view the information from each view (Figure 10.7).

Fuse-first multi-view tracking approaches are characterized by the features used and by the strategy for the computation of the common view (Figure 10.8). The features extracted can be the feet location of people [5], the silhouette centroid [19], the change segmentation mask [22], the foreground pixels or the whole motion segmentation likelihood [27].

Note that although fuse-first methods involve one tracking step only, they may involve multiple *detection* steps: (i) in each camera view, before fusion and (ii) on the common view, after fusion. Furthermore, as the fusion involves triangulation of noisy information, this can result in a larger number of solutions (i.e., candidate targets) than desired. To address this type of data and to reduce the overall complexity of the tracker, simultaneous detection and tracking can be performed that does not require a detection step (Figure 10.9). The various aspects of these multi-view tracking techniques are discussed in this section. A summary of the state-of-the-art of fuse-first multi-view tracking approaches is shown in Table 10.2.

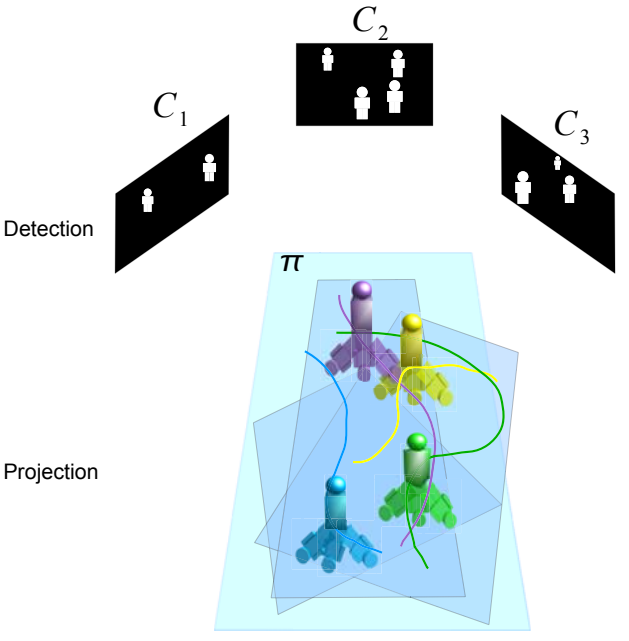


Fig. 10.7 Illustration of the fuse-first approach.

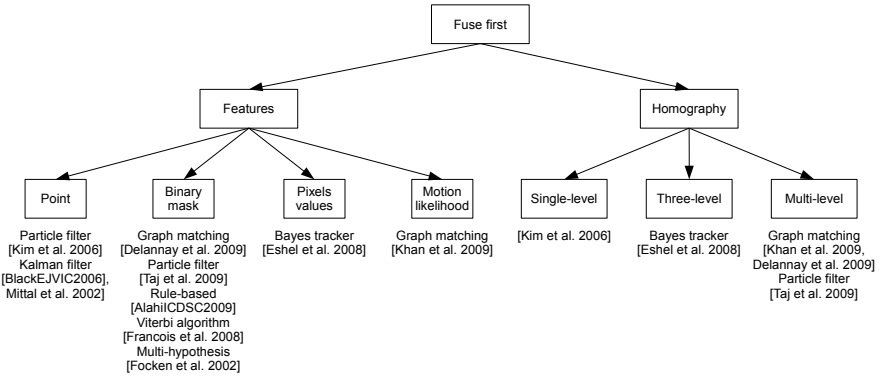


Fig. 10.8 Fuse-first multi-view tracking approaches: features and homography.

10.5.1 Detection-Based Tracking

Detection-based trackers first localize objects on the common view and then track them. Target localization (detection) can be performed by thresholding [21, 23] or by quantizing the top view into a grid such that each sub-area can only contain one target. A dictionary of atoms modeling the presence of an object at a given

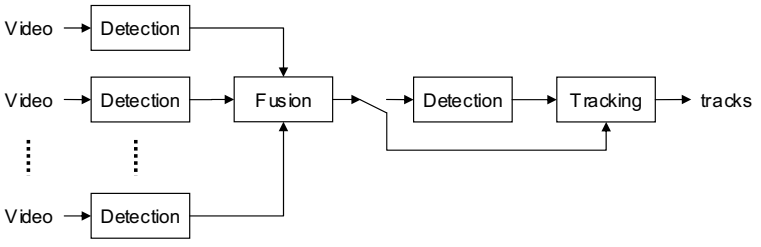


Fig. 10.9 Generic block diagram of fuse-first multi-view tracking algorithms. The switch differentiates between detection-based trackers and simultaneous detection & tracking algorithms that do not require the detection step after fusion.

Table 10.2 Fuse-first multi-view tracking algorithms (MT: Multi-target tracking; MHPT: Multi-Hypothesis Probabilistic Tracker; M = Manual; A = Automatic)

Ref.	Features	Tracker	Calib.	MT
[28]	color and motion	Viterbi algorithm	M	Yes
[5]	person vertical axis, ground position	Particle filter	M	Yes
[20]	feet position	Kalman filter	M	Yes
[19]	color histogram, bounding box, centroid	MHPT	M	Yes
[21]	head position	Bayes tracker	M	Yes
[16]	multiple planes occupancy map	Minimum graph cut	A	Yes
[27]	field of view lines	not mentioned/any	M	NA
[22]	foreground mask	Particle filter	M	Yes
[29]	foreground mask	Rule-based	M	No
[23]	foreground mask	Graph cut	M	Yes
[30]	2D position	Kalman filter	M	No

location in a view can then be used to identify if the position in the quantized top-view grid contains a target [29]. When the top view is composed of projected points representing target centroids or feet locations, all the non-zero values can be used as candidate locations [30]. A ray can be drawn from the center of projection of each camera through the centroid of foreground regions from that camera. The intersection of these rays can then be used as target location, which can be tracked using Multi-hypothesis Probabilistic Tracker (MHPT) [19]. Similarly, the vertical axis of the target across views can be mapped on the top-view plane and their intersection point on the ground can be used as the feet location of the target on the top view [5] (Fig 10.10). Contrary to [12], in [5] target feet locations (obtained through vertical axis lines) are not tracked in each camera view but detected and tracked only once on the common view. The detection step involves associating multiple projected points to the same target by using a threshold on the inter-point distance. The top-view feet locations can then be tracked using a single-view tracker such as particle filter. The thresholding on inter-point distance for associating multiple projected feet locations belonging to the same target can be eliminated by using a Gaussian kernel

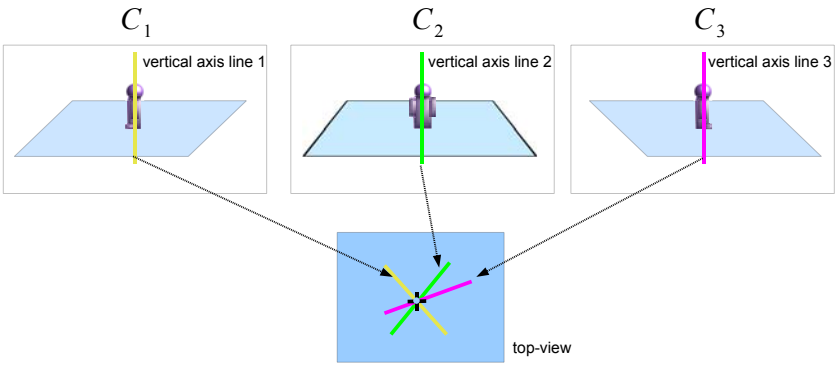


Fig. 10.10 Illustration of the target vertical axis intersection on the top view.

for a single image pair [20]. This results in a common plane that is similar to the one generated using foreground masks (Equation 10.4, Equation 10.6) or likelihood maps (Equation 10.8).

When the common view is based on foreground masks, object segmentation can also be performed by thresholding, thus resulting in a large number of points for each target. These points need to be grouped to obtain the target location. The grouping can be performed using K-means, Mixture of Gaussians or Mean-shift [31, 32]. The mean of these clusters represents the target location, which can be tracked using single-view trackers such as multiple single target Kalman filter [20]. Color and motion information can also be used in the common view with a generative model to explicitly handle complex occlusions and interactions between objects [28]. The tracking of each object can be performed using the Viterbi algorithm. A greedy approach that makes the locally optimal choice at each stage can be used to avoid the combinatorial explosion of the computational cost due to joint posteriors. Unlike approaches that perform state estimation using frame-to-frame correspondence only, this method computes global optima of scores summed over several frames, thus making it more robust to persistent and prolonged occlusions. However, this approach can only process a batch of frames at a time and hence the results are delayed.

To further improve the effectiveness of tracking in the fused domain, multi-level homography can be used [24] (see Section 10.3.2). Head detection can be performed by thresholding the variance map (Equation 10.4) and by employing floor-level homographic projections. Finally, the candidate head-top positions can be estimated by clustering with double threshold hysteresis. Note that head tracking requires the cameras to be mounted at a significant height so that the heads are fully visible. The number of homography levels can be increased to further improve the localization information [16], at an additional computational cost. The localization information can also be improved by projecting the motion segmentation likelihood values and obtaining the mask by taking the product of the values from multiple views (Equation 10.8). The foreground likelihood probabilities from each plane of each

view at each time can be projected onto the corresponding plane of the common-view to obtain a 4D spatio-temporal occupancy map. The minimum graph cut procedure can be applied with alpha-expansion to segment targets. Trajectory segmentation can then be performed using graph cut. Although this approach shows promising results, it is computationally very expensive as it requires obtaining a 4D occupancy map before applying the minimum cut procedure.

To reduce the computational cost and to obtain an on-line solution, multiple homography planes can be collapsed on the ground plane [23]. Similar thresholding and clustering can be performed to localize targets followed by graph matching to obtain the tracks.

The thresholding step to localize targets is a bottleneck in most detection-based trackers. Furthermore, due to parallax error (Figure 10.4), false peaks can be selected as candidate target locations. These false peaks can be filtered using heuristics on size and speed [25]. A better alternative is to perform tracking without applying the detection step by using simultaneous detection and tracking via track-before-detect.

10.5.2 Track-Before-Detect

Track-before-detect (TBD) is a Bayesian approach that extends the target state with the signal intensity and evaluates each image segment against a certain dynamic model. As the target intensity along with its dynamics follow a statistical model, this approach allows us to track targets with lower signal strength, without applying an additional detection step. The state estimation can then be performed using particle filtering [22].

To avoid an explicit target localization step, in track-before-detect the entire input signal is considered as a measurement. This measurement is a highly non-linear function of the target state and can be solved either by discretization of the state [34]

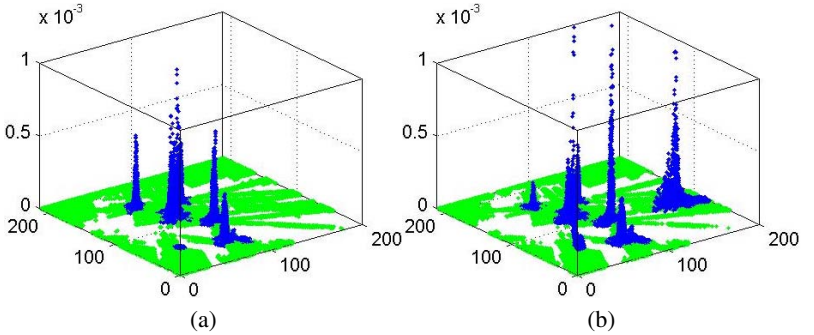


Fig. 10.11 Example of particle weights and positions. (a) Without multi-target update (one target has very small weights and another one is missing); (b) with multi-target update. As the weights for weak targets are very low, without the multi-target update strategy, lost tracks are possible.

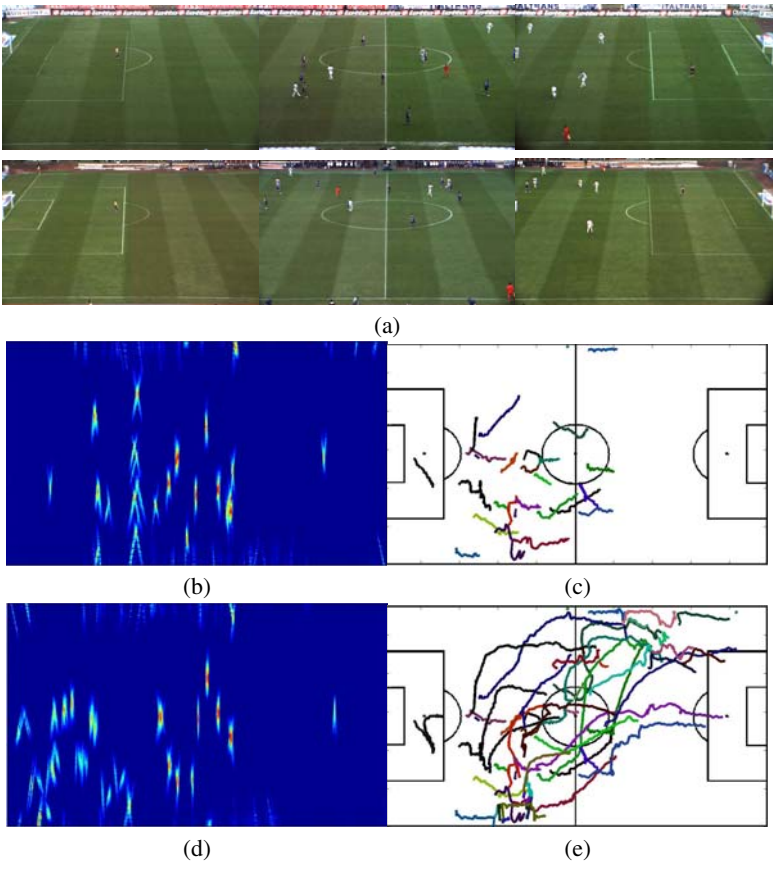


Fig. 10.12 Multi-view tracking on the top view on frames 160 and 500 of ISSIA dataset. (a) Original frames from each view. (b,d) Top view after fusion. (c,e) Tracks generated with multi-target track-before-detect.

or by non-linear state estimation techniques (e.g., particle filtering [35]), which are less computationally expensive.

In track-before-detect multi-view tracking, the common view can be based on the foreground likelihood (Equation 10.8) or the binary mask (Equation 10.6, Equation 10.4). The single-target multi-view track-before-detect particle filter can be modified for multiple targets by incorporating particle clustering [22]. The cluster information allows normalizing weights per target/cluster, thus facilitating tracking weak and new born targets.

Figure 10.11 shows a comparison between the evolution of particle weights with and without the cluster-based update strategy. It can be seen that without the multi-target update strategy (Figure 10.11(a)), a target is lost while another has a very low weight that will cause that target to be lost in the subsequent frame.

The particles can be clustered using K-means, Mixture of Gaussians or Mean-shift (MS) [36]. If the total number of targets is not known, a nonparametric clustering technique that does not require prior knowledge of the number of clusters, such as MS, can be used. MS climbs the gradient of a probability distribution to find the nearest dominant mode or peak and does not impose constraints on the shape of the clusters.

An example of tracking results obtained with the multi-target particle filtering track-before-detect (MT-PF-TBD) on the ISSIA[2] dataset is shown in Figure 10.12(c,e). The bandwidth chosen for MS is $h = 5$, which is appropriate for clustering particles generated around a target that is affected by blurring; 3000 particles per target are used. It can be seen that most targets are tracked over the entire scene, the exception being the goalkeeper on the left corner of the field. This target is not tracked initially (Figure 10.12(c)) despite being represented with significant information (Figure 10.12(b)) as he was static and hence not following the expected motion model. The prediction resulted in moving all particles away from the target. The corresponding track is generated when he starts moving during the attack on the goal (Figure 10.12(d-e)).

10.6 Conclusions

This Chapter discussed and classified techniques for tracking in multiple cameras with partially overlapping fields of view. The Chapter covers the two major groups of multi-view tracking algorithms, namely track-first and fuse-first approaches. Track-first approaches employ tracking in each view as well as on the common view. Trackers in each view can also collaborate with each other to improve the target estimates. Contrary to track-first methods, fuse-first approaches defer tracking until the fusion of target localization information on the common view. Tracking is then performed only once on the common view using multiple single-target trackers or multi-target trackers. Tracking on the common view can be based on detections (when targets are first localized prior to tracking) or on simultaneous detection and tracking. In this context, the Chapter has presented a track-before-detect multi-target particle filter tracker where only pixels following a certain dynamic model are tracked, without any explicit detection mechanisms. This approach not only eliminates the detection step after data fusion, but also helps reducing false positives due to noise.

References

1. Taj, M., Cavallaro, A.: Audio-assisted trajectory estimation in non-overlapping multi-camera networks. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Taipei, Taiwan (2009)

[2] Raw videos courtesy of Institute of Intelligent Systems for Automation - C.N.R., Bari, IT. http://www.issia.cnr.it, last accessed: 26 June, 2008

2. Nettleton, E., Durrant-Whyte, H., Sukkarieh, S.: A robust architecture for decentralised data fusion. In: Proceedings of the International Conference on Advanced Robotics, Coimbra, PT (2003)
3. Medeiros, H., Park, J., Kak, A.C.: Distributed object tracking using a cluster-based kalman filter in wireless camera networks. Journal of Selected Topics In Signal Processing (2008)
4. Soto, C., Song, B., Roy-Chowdhury, A.: Distributed multi-target tracking in a self-configuring camera network. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1486–1493 (2009)
5. Kim, K., Davis, L.S.: Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 98–109. Springer, Heidelberg (2006)
6. Javed, O., Shafique, K., Shah, M.: Appearance modeling for tracking in multiple non-overlapping cameras. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 26–33 (2005)
7. Morariu, V., Camps, O.: Modeling correspondences for multi-camera tracking using non-linear manifold learning and target dynamics. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2006)
8. Morioka, K., Mao, X., Hashimoto, H.: Global color model based object matching in the multi-camera environment. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 2644–2649 (2006)
9. Cai, Q., Aggarwal, J.: Tracking human motion in structured environments using a distributed-camera system. Transaction on Pattern Analysis and Machine Intelligence 21, 1241–1247 (1999)
10. Black, J., Ellis, T., Rosin, P.: Multi view image surveillance and tracking. In: Proceedings of the International Workshop on Motion and Video Computing (2002)
11. Anjum, N., Cavallaro, A.: Trajectory association and fusion across partially overlapping cameras. In: Proceedings of the International Conference on Advanced Video and Signal Based Surveillance (2009)
12. Du, W., Piater, J.: Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 365–374. Springer, Heidelberg (2007)
13. Pham, N., Huang, W., Ong, S.: Probability hypothesis density approach for multi-camera multi-object tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 875–884. Springer, Heidelberg (2007)
14. Qu, W., Schonfeld, D., Mohamed, M.: Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. EURASIP Journal on Applied Signal Processing, 21 (2007)
15. Kayumbi, G., Cavallaro, A.: Multiview trajectory mapping using homography with lens distortion correction. EURASIP Journal on Image and Video Processing (2008)
16. Khan, S., Shah, M.: Tracking multiple occluding people by localizing on multiple scene planes. Trans. on Pattern Analysis and Machine Intelligence 31, 505–519 (2009)
17. Stauffer, C., Tieu, K.: Automated multi-camera planar tracking correspondence modeling. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2003)
18. del Rincon, J.M., Herrero-Jaraba, J.E., Gmez, J.R., Orrite-Urunuela, C.: Automatic left luggage detection and tracking using multi-camera ukf. In: Proceedings of the International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 51–58 (2006)
19. Focken, D., Stiefelhagen, R.: Towards vision-based 3-d people tracking in a smart room. In: Proceedings of the International Conference on Multimodal Interfaces (2002)

20. Mittal, A., Davis, L.S.: M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 18–33. Springer, Heidelberg (2002)
21. Eshel, R., Moses, Y.: Homography based multiple camera detection and tracking of people in a dense crowd. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2008)
22. Taj, M., Cavallaro, A.: Multi-camera track-before-detect. In: Proceedings of the International Conference on Distributed Smart Cameras (2009)
23. Delannay, D., Danhier, N., Vleeschouwer, C.D.: Detection and recognition of sports (wo)man from multiple views. In: Proceedings of the International Conference on Distributed Smart Cameras (2009)
24. Khan, S.M., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 133–146. Springer, Heidelberg (2006)
25. Yang, D., Gonzalez-Banos, H., Guibas, L.: Counting people in crowds with a real-time network of simple image sensors. In: Proceedings of the International Conference on Computer Vision, pp. 122–129 (2003)
26. Monier, E., Wilhelm, P., Rckert, U.: Multi camera based tracking of indoor team. In: Proceedings of the International Conference on Distributed Smart Cameras (2009)
27. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. Transaction on Pattern Analysis and Machine Intelligence 25, 1355–1360 (2003)
28. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. Transaction on Pattern Analysis and Machine Intelligence, 30, 267–282 (2008)
29. Alahi, A., Boursier, Y., Jacquesy, L., Vandergheynst, P.: Sport players detection and tracking with a mixed network of planar and omnidirectional cameras. In: Proceedings of the International Conference on Distributed Smart Cameras (2009)
30. Black, J., Ellis, T.: Multi camera image tracking. Elsevier Journal of Image and Vision Computing 24, 1256–1267 (2006)
31. Jain, A., Flynn, M.M.P.: Data clustering: A review. ACM Computing Surveys 31, 264–323 (1999)
32. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. Transaction on Pattern Analysis and Machine Intelligence 25, 564–577 (2003)
33. Czyz, J., Ristic, B., Macq, B.: A particle filter for joint detection and tracking of color objects. Elsevier Journal of Image and Vision Computing 25, 1271–1281 (2006)
34. Bruno, M.G.S., Moura, J.M.F.: Multiframe detector/tracker: optimal performance. Transaction on Aerospace and Electronic Systems 37, 925–945 (2001)
35. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. Transaction on Signal Processing 50, 174–188 (2002)
36. Comaniciu, D., Meer, P.: Distribution free decomposition of multivariate data. Transaction on Pattern Analysis and Machine Intelligence 2, 22–30 (1999)