136. Multi-target tracking in video

136. Multi-target tracking in video

Fabio Poiesi and Andrea Cavallaro

November 12, 2012

Contents

Chapter 1

136. Multi-target tracking in video

Abstract

Multi-target tracking in video helps in gathering information from motion patterns to describe behaviors (e.g. sport team formations), to detect events of interest (e.g. crossing streets in forbidden locations) and to facilitate content retrieval (e.g. automatic highlights generation). Several challenges affect multi-target tracking, including color and shape similarities, occlusions and abrupt motion variations. We define a generic flow diagram that we use to discuss and compare the main stages of multi-target trackers, namely feature extraction, target prediction, localization or association, and post-processing. Trackers may also learn about the environment they operate in (contextual information) and update the target model they use in order to enhance the localization task. We finally summarize the properties of the surveyed multi-target trackers and introduce open research problems in video tracking research.

1.1 Introduction

The demand for the automated analysis of the behavior of people, animals and moving objects such as vehicles has grown considerably in the past years. For example, systems for the recognition of human actions and the detection of abnormal behaviors are key to support surveillance tasks [15]. To this end, video trackers enable motion pattern analysis of single and multiple targets [89]. Single-target trackers help analyzing motion patterns and behaviors of individuals, separately. Multi-target trackers help quantifying target interactions and comparing motion patterns of different objects simultaneously (Fig. 1.1). Surveillance systems (Fig. 1.2a) use trackers to monitor behaviors [77], to follow selected people and to recognize them in the view of other cameras [12, 57]. The analysis of collective and individual trajectories can be exploited to recognize abnormal behaviors in crowds [68]. Trajectories can be used to recognize interactions among humans [75] and to monitor the activity of people in order to analyze social behaviors [26]; to observe interactions among objects and humans, to help studying collaborative behaviors in meeting rooms [84], or to monitor the position of people with respect to abandoned objects [77]. Tracking is also used in video-based sport analysis (Fig. 1.2b) for automatic summarization [19] and statistics gathering [32, 76]. In traffic scenes, tracking using fixed or airborne cameras [90] helps to automatically detect unlawful U-turns, vehicles driving in the wrong direction, people crossing roads [33] (Fig. 1.2c) and to collect statistics on typical and atypical behaviors of vehicle and pedestrian flows [4].

This chapter is organized as follows. In Sec. 1.2, we introduce the general framework of multi-target trackers and formulate the multi-target tracking problem. In Sec. 1.3, we discuss real-world challenges for video-based tracking. Section 1.4 describes features that can be extracted from a video to enable tracking. In Sec. 1.5, we discuss prediction models used to estimate the location of targets. Section 1.6 describes sequential estimation and batch association to generate tracks, whereas in Sec. 1.7, we analyze different methods for track initialization and track termination. Next, Sec. 1.8 describes the use of contextual information to facilitate multi-target tracking. Finally, in Sec. 1.9, we summarize the chapter and discuss open research problems.

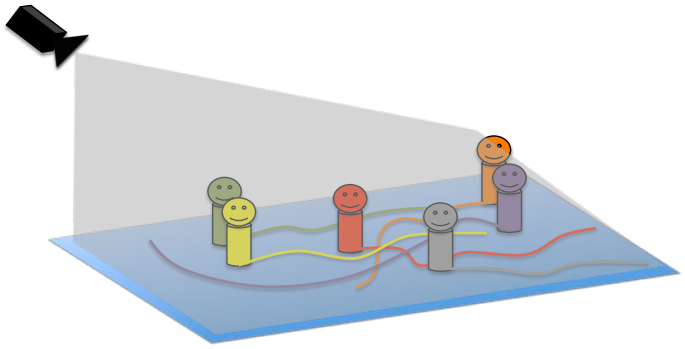
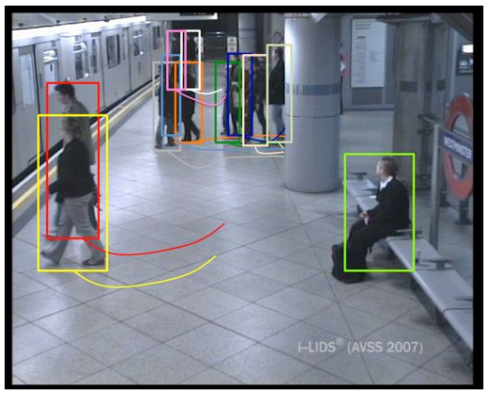
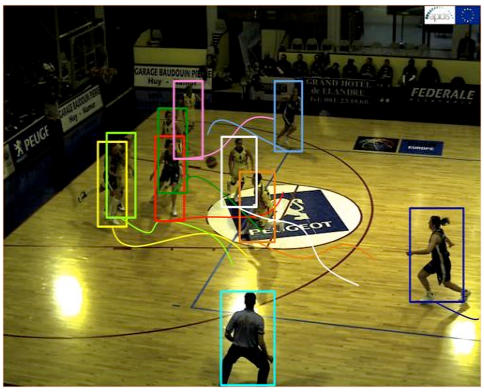


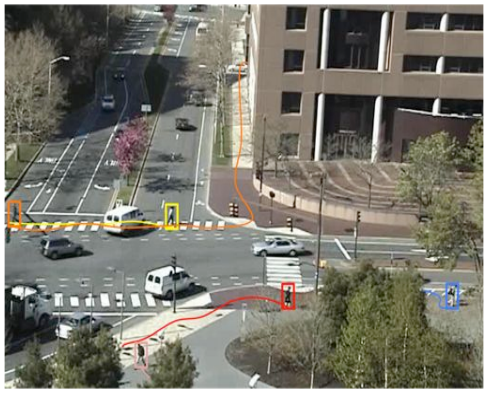
Figure 1.1: Example of multi-target tracking in video. The colored tracks are associated to the respective targets.



(a)



(b)



(c)

Figure 1.2: Examples of video-based applications that benefit from multi-target tracking: (a) surveillance (image from iLids dataset for AVSS 2007 [36]); (b) sport analysis (image from APIDIS dataset [5]); (c) automatic pedestrian flow monitoring (image from MIT Traffic dataset [85]).

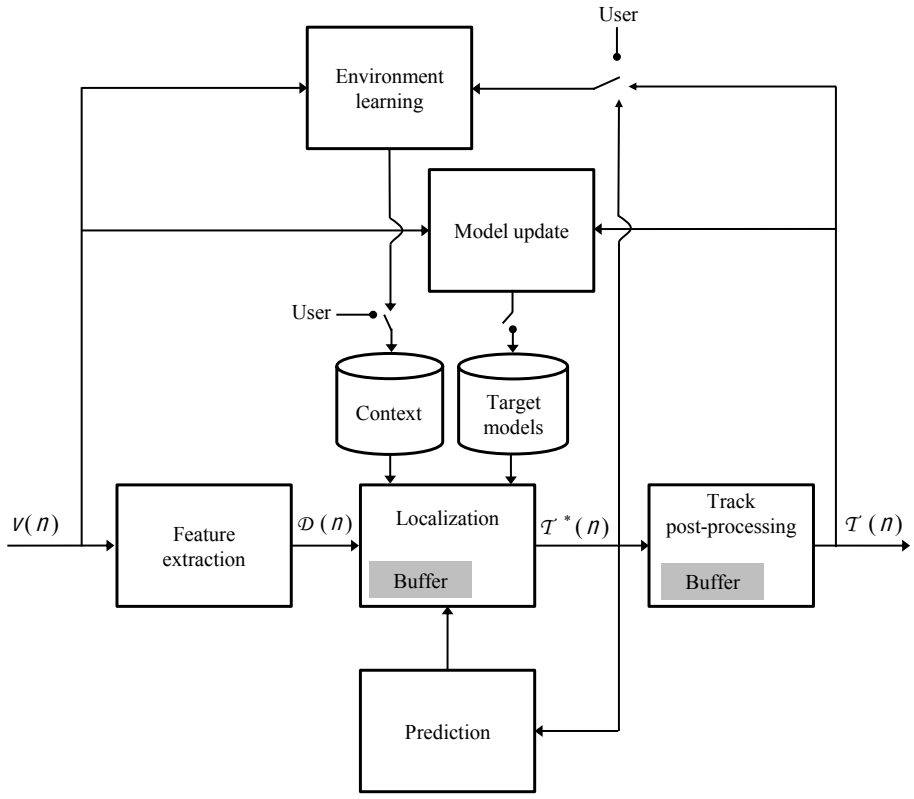


Figure 1.3: Block diagram of a tracker with sequential localization. The buffer accumulates measurements to allow processing over temporal windows. The localization stage can be externally initialized or can use contextual information, such as a map of the scene, and target model. The track post-processing stage improves the quality of the final result, for example by linking short tracks or by deleting spurious tracks.

1.2 Problem formulation

The goal of multi-target tracking in video is to generate accurate estimations of target trajectories (tracks) within the field of view of a camera. A tracker can be divided into four main stages (Fig. 1.3): feature extraction, localization or association (which exploits features to identify the position of the targets on the image plane), prediction (which models the motion of targets to predict their future locations), and track post-processing. In order to enhance the localization task, the output tracks can be used to extract contextual information by learning the environment [51] and by updating the model of the targets [41, 70].

Let $\mathcal{V} = \{v(n)\}_{n=1}^{N}$ be a video sequence, where $v(n)$ is the n^{th} frame and N is the total number of frames. The feature extraction stage generates at time n a set $\mathcal{D}(n)$ of filtered features

$$\mathcal{D}(n) = \{\mathbf{d}_h(n) : n, h \in \mathbb{N}\}, \tag{1.1}$$

where

$$\mathbf{d}_h(n) = [u_h(n)\ v_h(n)\ \mathcal{I}_h(n)]^T \tag{1.2}$$

is the h^{th} feature, $u_h(n)$ and $v_h(n)$ are the positions with respect to the horizontal and vertical axes, respectively; $\mathcal{I}_h(n) \in \mathbb{R}_{[0,1]}$ is a scalar value between 0 and 1 that indicates the confidence of that feature representing a target, and T is the transpose of a matrix. Features belonging to the same targets are then linked over time in order to estimate tracks. Generally, target localization or association are defined as a function $f(\cdot)$ such that

$$\tau_m(n) = f(\mathcal{D}(n-\gamma_1), ..., \mathcal{D}(n+\gamma_2)), \tag{1.3}$$

where $\tau_m(n)$ is the m^{th} track up to frame n and $\mathcal{D}(n-\gamma_1), ..., \mathcal{D}(n+\gamma_2)$ are the input features measured in the interval $[n-\gamma_1, n+\gamma_2]$ (measurements), where $\gamma_1, \gamma_2 \in \mathbb{N}_0$. The track $\tau_m(n)$ belongs to the set of tracks $\mathcal{T} = \{\tau_m\}_{m=1}^{M}$, where M is the total number of tracks computed within the sequence $\mathcal{V}$. The track of a generic target m is a time series

$$\tau_m = \{\mathbf{x}_m(n) : 1 \leq n \leq N\}, \tag{1.4}$$

where $\mathbf{x}_m(n) \in \mathbb{R}^d$ is the state of the target at frame n and d represents the dimension of the state. The information encoded in the state $\mathbf{x}_m(n)$ is used to describe the status of the target at frame n. The definition of $\mathbf{x}_m(n)$ is application-dependent. For example, $\mathbf{x}_m(n)$ may encode the position and velocity of the target [65], or also size information, such as width and height of the target [17]. The simplest representation of a target onto the image plane is its 2D-position ($d = 2$),

$$\mathbf{x}_m(n) = [x_m(n)\ y_m(n)]^T, \tag{1.5}$$

where $x_m(n)$ and $y_m(n)$ represent the target position on the horizontal and the vertical axes, respectively.

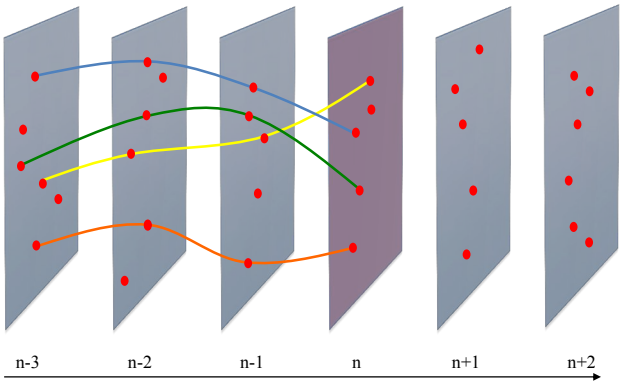
Some approaches formulate the problem of *simultaneously tracking M targets* as a problem of single-target tracking, M times. The target-tracker association is performed by an external algorithm that guarantees that one tracker is exclusively associated to a target [30]. Alternatively, multi-target tracking can be formulated as the problem of jointly tracking all the targets by using a single tracker [17]. When the number of targets increases, maintaining the identities of all the tracks correctly associated to the targets becomes challenging. Interactions among neighboring targets can also be modeled [30, 39].

To improve target localization, one can calculate the state $\mathbf{x}_m(n)$ based on the predicted state $\hat{\mathbf{x}}_m(n)$ and the current measurements [65]. The predicted state $\hat{\mathbf{x}}_m(n)$ is calculated using a function $h(\cdot)$ applied on the state at the previous frame $n-1$, such that

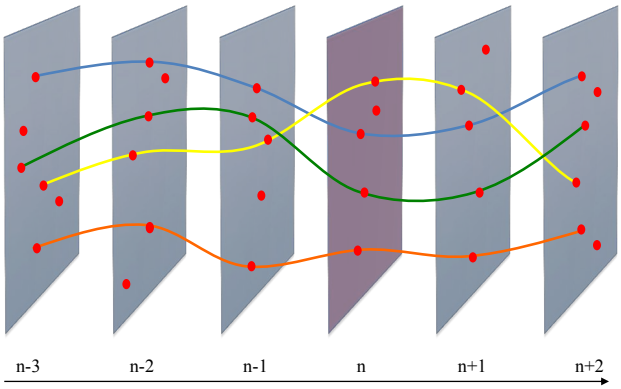
$$\hat{\mathbf{x}}_m(n) = h(\mathbf{x}_m(n-1)). \tag{1.6}$$

The function $h(\cdot)$ is also known as motion model or evolution model. The motion model is used to draw state hypotheses from the current frame to the next, mostly using kinematic models [65]. These hypotheses are further validated using features extracted from the current image frame. Hence, the state $\mathbf{x}_m(n)$ is estimated using the previous state $\mathbf{x}_m(n-1)$ and the measurements from the image at time n. Motion models can be either pre-learned [2, 30, 41, 47, 66] or fixed [1, 8, 10, 12, 17, 31, 32, 59, 73, 83, 88, 92].

Trackers can be causal or non-causal filters. *Causal trackers* ($\gamma_1 > 0, \gamma_2 = 0$) only use features extracted from the past and the current time step n to estimate tracks (see Fig. 1.4a). Causal trackers, such as tracking methods based on particle filtering [6] and Markov Chain Monte Carlo (MCMC) [3], are used for time-critical applications. *Non-causal trackers* ($\gamma_1 \geq 0, \gamma_2 > 0$) use also future time steps, thus resulting in a delayed decision (see Fig. 1.4b). Non-causal tracking [9, 34, 90, 91] is typically formulated as a global optimization problem to retrieve target tracks throughout the video sequence [64]: the candidate target locations for the whole sequence [14] are obtained at the feature extraction stage and are then linked together using optimization processes [34, 48]. Motion models are implicitly included into the optimization algorithm and they are commonly expressed as constant velocity models [14]. Non-causal methods can be divided into



(a)



(b)

Figure 1.4: Causal and non-causal multi-target tracking: (a) causal trackers operate using measurements from the current and past instants; (b) non-causal trackers generate the results using past, current and future observations.

two categories: (i) methods that iteratively compute long tracks by associating time-independent features and (ii) methods that build long tracks in multiple steps, by extracting short-term track either with causal or sub-optimal association trackers and then, by associating shorter tracks (i.e. tracklets) into longer tracks. Examples of non-causal trackers [62] include Detection Association Trackers (DAT), such as Multiple Hypothesis Tracking (MHT) [64] and Joint Probabilistic Detection Association (JPDA) [7].

(a) (b)

Figure 1.5: Example of color variations of a target due to illumination differences: (a) in a shop; (b) in a corridor. Images from CAVIAR dataset [13].

Figure 1.6: Example of clutter: the person to be detected is not clearly distinguishable due to appearance similarity with the background and with other objects (e.g. a mannequin). Image from CAVIAR dataset [13].

1.3 Challenges

The challenges a tracker may face are due to color similarities among objects, illumination changes, pose variations, occlusions, various noise components, abrupt or unpredicted motion variations, and the density of targets in the scene.

Color similarities can mislead the target-background and the target-target discrimination. When another region in the image has similar color to that of a target, then a track can be lost. Similarly, when targets with similar color move close to each other, their identities can be swapped [44, 87]. *Illumination* changes caused, for example, by different light sources (Fig. 1.5) lead to color variations that can induce target losses. This problem can be addressed by using illumination invariant features or by updating the color model of the targets [70].

Shape similarities can also generate ambiguities (Fig. 1.6) between a target and an object in the background, or among similar targets [86]. Examples include people tracking when the shape is encoded as the head–and–shoulder or full–body outline [23]. *Pose* variations leading to shape changes (Fig. 1.7) require a tracker to be capable of adapting the corresponding appearance models to avoid track inaccuracies or losses [53].

When *occlusions* happen (Fig. 1.8), the only data available to the tracker are the measured target dy-

(a) (b)

Figure 1.7: Example of shape change due to pose variation: (a) front view; (b) side-rear view. Images from CAVIAR dataset [13].

Figure 1.8: Example of a largely occluded person (man with the black jumper). Image from CAVIAR dataset [13].

namics and the appearance features before (and after) the occlusion itself [12, 87]. Using this information, a tracker can estimate the likely location of a target by interpolating spatio-temporal data when the target is not observable.

Noise components can be introduced during video acquisition or compression, and may lead to corrupted measurements that generate unreliable features for the estimation of the target locations.

Motion-related challenges can be due to abrupt variations (e.g. sudden accelerations) or unusual dynamics (e.g. deviations for a predictable path to avoid obstacles). Most tracking algorithms rely on prior models for motion prediction [12]. These models are mainly linear with additional terms that represent small variations as noise components [65].

Finally, the difficulty of tracking depends on the *density* of targets in the scene. In the case of people tracking, when the crowd motion tend to be coherent in one direction and does not vary over time because of the high spatial density of people, crowded scenes are defined as *structured*. In *unstructured* crowded scenes, instead, groups of people may move simultaneously in different directions [66]. Successful trackers rely on part-based detectors and perform non-causal tracking using target re-identification [44].

1.4 Feature extraction

The tracks can be estimated either by extracting features in each frame and linking them over time, or by extracting features at initialization (i.e. when an object appears in the scene) and then by letting the tracker perform the subsequent location hypotheses and confirming them over time using the newly extracted features. Example of features are discussed below and summarized in Tab. 1.1.

Target candidate locations can be defined using *color* histograms [28, 43, 44, 62, 87, 91]. Color histograms are used to distinguish targets over time while addressing the challenges of targets with similar color. In order to reduce sensitivity to light variations, histograms are generally quantized with 8 bins per RGB channel [43].

Detections indicating candidate targets, represented as vectors or binary maps [10, 86], can be provided by cascade classifiers [82, 94] or by multi-valued maps [73] representing the confidence of having targets in specific locations. The latter case allows one to deal with targets with low signal–to–noise ratios [65]. When the scene background is fixed, it is possible to detect moving targets by calculating the difference between the current frame and a reference background frame [61]. One can use a simple weighted frame difference between the actual and the background frame [8, 15], or a mixture of Gaussians [92] where each component of the mixture belongs to a color channel (e.g. three components for RGB). In this case, the mixture of Gaussians is learned on the background and the probability of each pixel of a new incoming frame being considered a part of background or of target is then calculated.

Shapes can be used to represent targets, for example, in the form of Histograms of Oriented Gradients (HOG) [18]. This representation is popular for describing heads [10] and bodies [12, 73]. Alternatively, *edgelets*, a large pool of short lines and curve segments (based on intensity gradients), can be used to represent human shapes [86]. This method employs descriptors for head, torso, leg and full body, which are combined to address the problem of occlusions. Similarly, *shapelets* are combinations of oriented gradient responses learned in a discriminative manner on local patches [69].

Targets can also be described with covariance matrices as *texture* descriptors. In this case, a dense model of covariance features (e.g. spatial location, intensity, higher-order derivatives) is used inside a detection area [80]. A target can be represented with several covariance descriptors of overlapping regions, where the best descriptors are determined with a greedy feature-selection algorithm combined with boosting. The covariance matrix descriptor is applied on image patches to characterize and distinguish targets [43, 44]. The Scale-Invariant Feature Transform (SIFT) [50] can be also used to capture texture characteristics in order to describe for example human torso regions [88].

The choice of the type of features provided to the localization or association stages is important regarding the use of a tracker with static or moving cameras [20, 45]. If the feature extraction relies only on target information, such as outline of targets [86], the tracker can be extended to moving camera applications [44]. Instead if the feature extraction relies on the background information for extracting candidate target locations (e.g. using background subtraction) and if the localization stage is highly dependent on this feature, major modifications to the localization stage are needed to extend the tracker from static to moving cameras [92]. An adaptation to moving cameras is also needed when contextual information (e.g. entry/exit points) is included in the localization stage [87].

1.5 Prediction

Predictive models generate target hypotheses that the localization stage validates using current measurements [52, 65]. Predictive models can use for example kinematic equations (e.g. constant velocity) [65] or

motion estimation models (e.g. [58]), and can involve a training phase of the target evolution [41]. Learning-based models mostly exploit a time interval at the beginning of a video sequence for training [2].

Particle filter algorithms [6] use autoregressive motion models for linearly predicting future target locations [1, 12, 17, 30, 31, 59, 73]. Equation 1.6 for a generic autoregressive motion model takes the following form:

$$\hat{\mathbf{x}}_m(n) = F\mathbf{x}_m(n-1) + \boldsymbol{\xi}(n-1), \tag{1.7}$$

where F is a $d \times d$ matrix defining the linear function $h(\cdot)$ and $\boldsymbol{\xi}(n-1)$ is random noise with a given distribution (e.g. Gaussian). Prediction models can be built using motion estimation algorithms [58] between consecutive frames. The resulting motion flow is exploited to build predictive models. A predictive motion model for consecutive features can be designed using a constant velocity model with the contribution of Kanade-Lucas-Tomasi (KLT) point tracks [10, 67, 79]. Specifically, the prediction state is defined as

$$\hat{\mathbf{x}}_m(n) = \mathbf{x}_m(n-1) + \gamma\hat{\mathbf{v}}_m(n-1), \tag{1.8}$$

where $\hat{\mathbf{v}}_m(n-1)$ is the velocity estimation coming from the KLT tracks in the frame prior the current state and γ is the time interval between the states where the velocity is calculated.

Mode-seeking trackers, such as the Mean-Shift tracker [16], follow neighboring modes of clusters generated with features extracted from the frames. Clusters are represented as modes and tracking is performed by seeking the closest mode in the subsequent frame [8]. The predicted location of the target in the subsequent frame lies in the area defined by a kernel that is dependent on the target position in the previous frame. Each mode displacement is therefore assumed to be smaller than the kernel size.

Learned models are used to improve performance when motion is predictable, for example, in case of high target density [2, 66]. Assuming that each target follows a coherent direction with respect to the other targets, it is possible to learn motion models and to include them into the tracker to help the prediction of target positions. For example, in crowded scenes, a set of motion constraints can be trained from the behavior of humans [2]. These motion constraints are properties retrieved from exit regions and dominant/common paths of people, influences generated by barriers or walls, and people behavior around the tracked person. The tracker may rely on a grid of particles over the image plane and tracking is performed by maximizing the transition probability of a particle from one cell to another. Hence, the transition probability is determined by two factors: (i) the color similarity between the current and the next location and (ii) the influence of the learned motion constraints in this location.

Scene dynamics can be learned using optical flow features [58] (i.e. position and velocity) and encoded according to a codebook, where each word of the vocabulary is associated to a specific dynamic [66]. The target location is computed using a weighted mean of the displacement of the observations based on the learned dynamics and the predicted displacement. Alternatively, time-varying dynamics of people across different spatial locations can be modeled using Hidden Markov Models (HMM) [41, 63]. The hidden states of the HMM encode possible motion patterns that are likely to be present at each spatial location.

1.6 Localization and association

Localization and association rely on the measurements coming from the feature extraction stage and validates feature similarities over time to estimate reliable tracks. The validation can be performed sequentially or as a batch process (Fig. 1.9). Sequential localization extracts tracks recursively. Bach processes are used when features are collected within a time interval and tracks are extracted by optimizing temporal links among features.

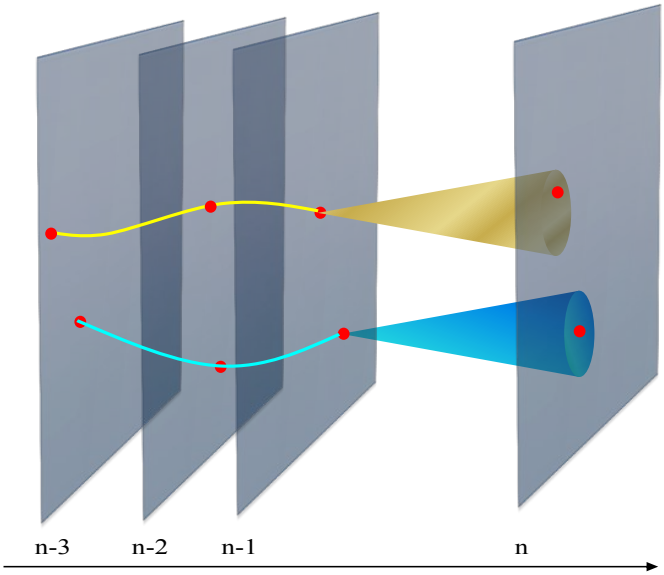
1.6.1 Sequential localization

Particle filter recursively finds targets using the Bayesian recursion for the sequential estimation of the target states over time [65]. The Bayesian recursion involves the estimation of the target state[1] $\mathbf{x}(n)$ calculated by constructing the posterior probability density function (pdf) using motion models (see Sec. 1.5) and measurements gathered from the current frame. The posterior pdf can be a multimodal distribution, where the modes of the distribution represent likely target locations. In order to make the Bayesian recursion computationally tractable, the posterior pdf is approximated with a Monte Carlo method [21], which consists of a set of random samples, or particles, drawn from the posterior pdf with associated weights.

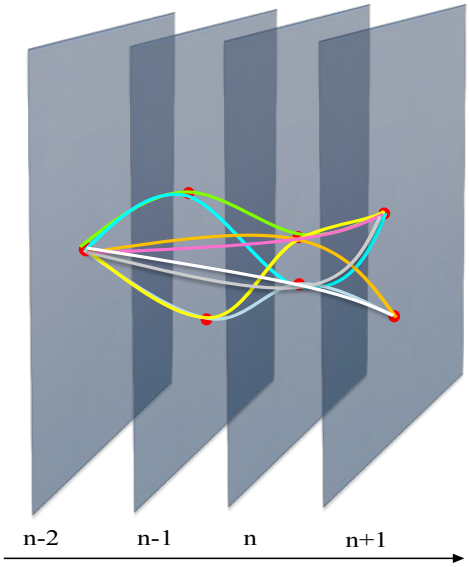
The extension from single to multi-target particle filter requires the size of the state to be made proportional to the number of targets, i.e. $\mathbf{x} \in \mathbb{R}^{d'}$ where $d' = d \cdot M$, with M being the number of targets [35]. Generally, when a single particle filter has to deal with multiple targets and the distribution of the states is represented with a mixture of distributions and one of the major problems is the maintenance of the multi-modality [81]. Hence, a mechanism based on AdaBoost for maintaining the multi-modality can be included into the tracker [59]. Alternatively, a baseline version of the particle filter is applied on confidence maps generated with a Cascade Confidence Filtering (CCF) which incorporates constraints on the size of the objects, on the preponderance of the background and on the smoothness of trajectories [73]. A feature extraction stage relying on geometric structures along with background filtering produces preliminary confidence maps for a certain time interval. Spurious features within this time interval are filtered out with a temporal smoothing method based on the Vessel filter [24]. The resulting confidence maps are used as observations for the multi-modal particle filter. Lastly, trackers can also rely on the detections generated, for example, by an AdaBoost classifier [59, 82].

Trackers can be composed of multiple particle filters, each of them operating on one target [1]. The communication among filters is performed with a heuristic method relying on the spatial locations of features. The feature–trajectory association is performed using the AdaBoost classifier confirming the tracking result for each trajectory. The assignment of the particle filters to each target can be performed by imposing a pseudo-independence among filters that is learned in a training phase [30]. Parameters include trained features into the weighting function of the filter, such as measures of the distance between target states and predicted locations. Features include the probability of the target being in a certain location with respect the predicted estimation, color similarity with respect to trained templates, dissimilarity with the background, penalty scores with target regions overlapping each other and neighboring targets. Furthermore, the Hungarian algorithm [42] can be used to associate particle filters to targets [88]. The assignment matrix is constructed using a Bayesian formulation among features and track states. The association between features and tracks can also be performed by greedy algorithms [12]. A single particle filter is employed for each target and, in order to discriminate the tracked targets, an on-line AdaBoost classifier is trained for each target against all the others. Each weak learner represents a feature computed for both positive and negative training images.

[1] The subscript m has been removed to generalize the problem.



(a)



(b)

Figure 1.9: Comparison between a sequential localization and a batch association approach. (a) Sequential localization uses predictive motion models to explore potential target areas. (b) Batch association generates track hypotheses (colored lines) that are selected based on an optimization procedure.

Random Finite Sets (RFS) can also be used along with the Bayesian formulation [52, 54]. RFS treat state and measurements as realizations of random variables and the Bayesian formulation with RFS can be approximated with Monte Carlo methods [31]. The feature extraction stage can be embedded into the tracker to define the appearance model and, like the motion model, it is defined a priori.

Markov Chain Monte Carlo (MCMC) methods can alternatively be used to drawn samples from posterior distributions, since with particle filter is very hard to deal with large number of targets and hence large state spaces. In fact, maintaining the multi-modality requires very precise mathematical methods [81] and the computational cost for handling high-dimensional state spaces is still prohibitive. For these reasons, in order to avoid expensive integration steps, MCMC methods have been introduced [3]. For example, Smith et al. [72] used MCMC to deal with 10-dimensional states. Moving humans can be represented with 3D models using camera calibration parameters after being detected via background subtraction. The information about their locations is employed to build a multi-person joint likelihood function and used to find person locations in consecutive frames, leading to a dimensionality of the space proportional to the number of persons in the scene [92]. Kalman filters are then used to build the posterior pdf for consecutive frames employing a fixed motion model describing persons at constant velocity and affected by Gaussian noise. Since a joint likelihood is used, which involves both discrete and continuous variables, MCMC is employed to sample from the posterior pdf and to obtain the estimation of the target states calculating the Maximum A Posteriori (MAP). Alternatively, track hypotheses can be extracted within a four-second window using Minimum Description Length (MDL) [10]. Features such as scale, location and motion computed with Kanade-Lucas-Tomasi (KLT) are associated over time using likelihood functions. A refinement stage relying on the likelihood functions is built to allow two types of modifications to track pairs, namely the move of certain features from one track to the other or the swapping of all the features belonging to both tracks at a chosen time instant. MCMC is then used to take decisions about the acceptance of such modifications and to confirm the final track decision.

Finally, sequential localization can be performed with ad-hoc methods, either based on thresholds or on combinations of different algorithms. For example, the link between two features can be defined by a probability (i.e. the link probability) calculated as a product of three independent affinities calculated from feature characteristics, such as position, size and appearance [34]. The final linking between two features is then confirmed by using a two-threshold strategy. The first threshold is used to check if the link probability is high enough; whereas the second threshold is used to determine if the affinity of any of their conflicting pairs is high enough.

1.6.2 Batch association

Features can be associated over time with a batch process through maximization algorithms applied on posterior probability, which quantifies the likelihood of the tracks given the set of features [91]. Let the set of H_d features $\mathcal{D} = \{\mathbf{d}_h\}_{h=1}^{H_d}$ be gathered from the video sequence and $\mathcal{T}^*$ be the set of track hypotheses obtained by associating features over time. The goal is to maximize the posterior probability of $\mathcal{T}^*$ given the set $\mathcal{D}$ (Fig. 1.9b), that is

$$\mathcal{T} = \arg\max_{\mathcal{T}^*} p(\mathcal{T}^*|\mathcal{D}) = \arg\max_{\mathcal{T}^*} p(\mathcal{D}|\mathcal{T}^*)p(\mathcal{T}^*) = \arg\max_{\mathcal{T}^*} \prod_{h=1}^{H_d} p(\mathbf{d}_h|\mathcal{T}^*)p(\mathcal{T}^*), \tag{1.9}$$

where $\mathcal{T} = \{\tau_m\}_{m=1}^{M}$ is the set of tracks and the likelihood probabilities are assumed to be conditionally independent given the hypothesis $\mathcal{T}^*$. Such maximization can be calculated with an iterative method that cycles through the sequence and finds optimal solutions between each consecutive frame pair [14]. The

two-frame optimal solutions are calculated by using 2D target locations with the Hungarian algorithm [42]. The iterative cycling method, similar to the Iterated Conditional Modes (ICM) algorithm [11], updates joint solutions of multiple variables in order to find stronger local optima, and the iterations continue until no further improvement are achieved.

There are methods to iteratively compute optimal tracks using the complete set of features [14], and methods that reduce the complexity of the problem by pruning negligible hypotheses and by finding sub-optimal solutions in multiple steps [34, 48].

An alternative method is formulated with a cost-flow network. Instead of using thresholds to link features [34], it is possible to use the algorithm for min-cost flow networks proposed by Goldberg [27]. Within this network, each flow is interpreted as a track of a single target and the cost of the flow corresponds to the log-likelihood of the link hypothesis. The log-likelihood linking is calculated by taking into account size, position, appearance and time gap of the features by considering independence among them. Features can be associated within a temporal window using Mean-Shift clustering [16] on the feature space [9]. For each cluster, which ideally represents a target, PCA is applied and the features are associated by considering the direction of the principal components. PCA allows one to represent the local trend in the data distribution and measure the reliability of the associated features.

Final tracks can be obtained by linking tracklets. This problem can be formulated as a joint problem of ranking and classification [48], by using HybridBoost, a combination of RankBoost [25] and AdaBoost [71]. The role of RankBoost is to build the tracklet affinity model considering relative preferences over any tracklet pairs as well as low values for those tracklet pairs that should not be associated. AdaBoost is composed of weak classifiers relying on a single type of features for tracklet affinity measurements, such as appearance, motion and frame gap between a tracklet pair.

An algorithm for optimal tracklet association (OLDAM) [43] uses a temporal shifting window for the online learning of discriminative appearance features. Positive samples are extracted within the same tracklet and collected for all the tracklets in a temporal window. Negative samples are collected by extracting features from tracklets not belonging to the same target and by taking into account their spatio-temporal properties. The model learning problem is formulated as a binary classification problem using AdaBoost. Affinity measurements of appearance features (i.e. color and HOG) are adopted in AdaBoost to learn weak classifiers. The predicted confidence output of AdaBoost is combined with motion and time features in order to compute the link probability between tracklets. OLDAM has been further improved with PIRMPT [44], which includes a method to automatically select the most discriminative features from each tracklet by an online learning method based on appearance descriptors. Such descriptors are used to create a target model for each tracklet and further employed to link consecutive tracklets [34]. Tracking improvements can be achieved with Explicit Occlusion Model (EOM), which includes in the tracking problem occlusion hypotheses [91]. The EOM method generates a set of occlusion hypotheses and constraints, and combines them with the input associations. This combination avoids errant associations due to large temporal gaps between the associated features.

In order to make the tracklet linking generic, methods shall be independent of feature extractors, for example, by employing an optimization process based on common affinity models along with social grouping behaviors [62]. The nonlinear equations used for the association can have terms approximated with Lagrange theory and solved using an iterative algorithm that employs the Hungarian algorithm and K-mean clustering [22].

1.7 Track initialization and termination

Initialization and termination of tracks are two important track management issues. The initialization for causal trackers can be performed automatically, i.e. a new track starts when new features are available and are not associated to any of the existing tracks. The initialization for non-causal trackers can be performed as for the causal trackers or with an implicit modality, i.e. when the initial location is associated to a track obtained as optimal solution. An alternative is manual track initialization, used for example in tag-and-track applications [2, 41].

Tracking methods performing batch association of features or tracklets [14, 43, 44, 87, 91] implicitly initialize and terminate tracks. In fact, when the optimal track solution is computed, the start and the end of each track are implicitly encoded into the solution. Instead, methods performing sequential localization need criteria for track initialization and termination.

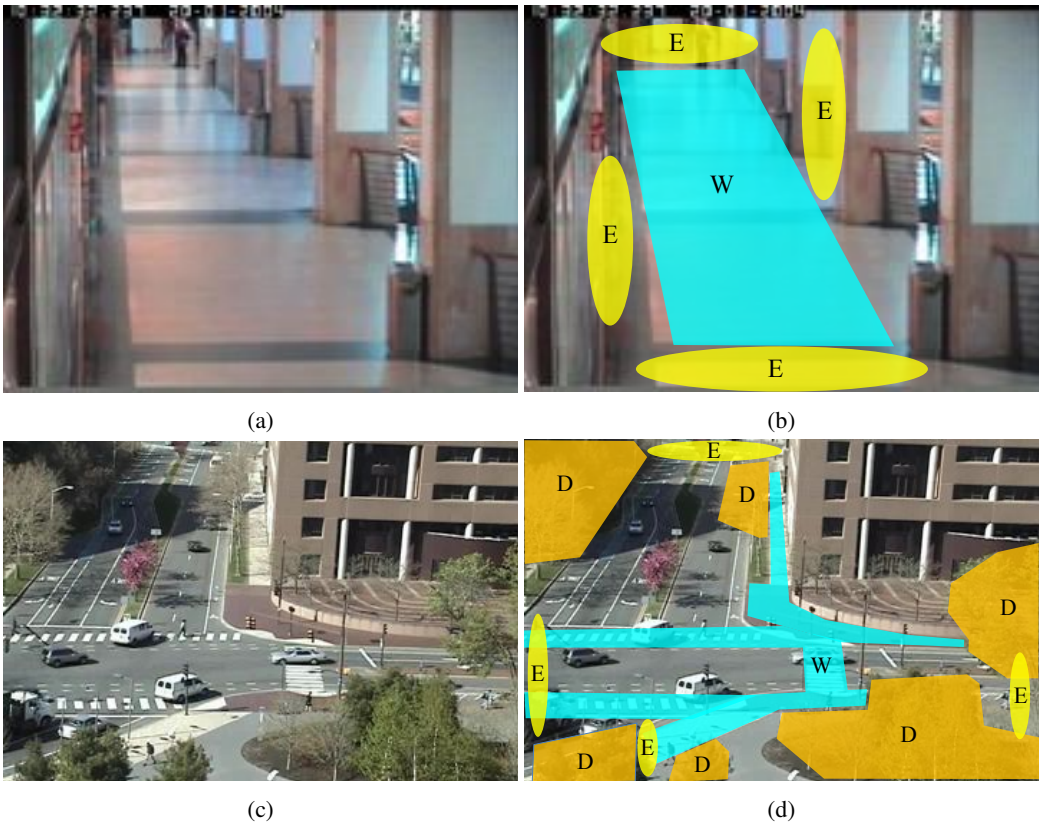
Trackers such as Track-Before-Detect based on particle filter [65] perform joint detection and tracking of targets without relying on any external mechanism for initialization or termination. The initialization and termination of tracks are embedded in the filter and modeled using a Markov chain [60], where the number of states corresponds to the number of targets in the scene. Alternatively, if the target states are represented as a collection of random variables that create a finite-set-valued state modeled with a multi-Bernoulli RFS [52], the tracker can handle track initialization and termination by relying on probabilities of target appearance and disappearance [31]. The RFS framework can handle a time-varying number of targets as well as missing and noisy features by employing for example the Probability Hypothesis Density (PHD) filter [55].

External mechanisms for track initialization or termination can be based on the extracted features [1, 12, 59, 82]. New tracks are initialized when none of the running trackers are associated to the localized targets [1, 59]. This process can be enhanced when multiple features are generated along the image borders, which is an indication on new incoming targets in the scene [12]. The termination of a track occurs when the tracker is unable to validate the features for a number of consecutive frames. Also, ad-hoc methods for initializing and terminating tracks can be used by implementing clustering strategies on the extracted features and comparing the number of clusters in the current frame with those in the previous frame [8].

1.8 Scene contextual information

Scene contextual information includes the knowledge of the scene background, occlusion areas, entry/exit regions, and dynamic textures (Fig. 1.10). Context is exploited to distinguish targets from clutter [51] and to improve initialization and termination of tracks [87]. For example, background information can be used to enhance the separability between target features and background features [73], or object-level information can be used to model spatio-temporal relationships in order to improve tracking in indoor scenarios [40, 49].

Contextual information can be extracted by learning the environment from user annotations or automatically from the output of the tracker. *User annotation* of entry/exit regions [87] may be required in order to provide the tracker with reliable contextual information. For example, detections of targets located in manually selected entry regions can be used to initialize tracks [12]. Alternatively, entry/exit regions, typical paths and stopping regions can be *automatically* extracted from long-term tracks [38, 56] or tracklets [87, 93]. In unstructured scenes, contextual information can be used to perform online learning of motion maps [87]. A motion map can be constructed by relying on entry/exit regions of the scene and by using motion patterns gathered from tracks. Entry/exit regions are used to draw likely target paths when reliable target features are collected. The learning of non-linear motion patterns is used to enhance the diversity among different



(a) (b)

(c) (d)

Figure 1.10: Examples of scene contextual information. Multi-target tracking can be enhanced by exploiting knowledge about the scene layout and objects such as trees that may occlude targets or may generate dynamic textures. Typical walking paths can also be used to narrow the search of human targets. Entry and exit regions can help track initialization and track termination. (Key: D: Dynamic textures; W: Walking path; E: Entry/exit regions). (a,b) Images from CAVIAR dataset [13], (c,d) images from MIT Traffic dataset [85].

track hypotheses, to improve the affinity estimations among extracted features and to build robust appearance models [87]. In structured scenes, scene context can be incorporated by automatically learning floor fields [2], which model directions of people on dominant paths and towards preferred exit regions, and to improve the motion prediction in these regions. Finally, with an *interactive* environment learning, models can be learned for clutter areas and for initialization areas. The clutter model improves the capability of the tracker to discard noisy measurements. The initialization model can reduce the delay of track initialization in locations where targets are likely to appear [51].

Scene contextual information is also modeled and used to improve tracking accuracy when linking tracklets [34, 87]. For example, the scene model (i.e. entry/exit regions and static occluders) projected on the ground plane with homography from the image plane [29] can be used to reduce track fragmentation and prevent identity switches of linked tracklets. Long-range trajectory association is performed using an Expectation-Maximization (EM) algorithm. The E-step estimates the scene model in terms of entry/exit regions with a Bayesian inference. Then these regions are used to specify initialization and termination of

Table 1.1: Taxonomic summary of multi-target trackers. Key: Ref: Reference; TI: Track initialization; TT: Track termination; BS: Background subtraction; RGB: Red Green Blue colorspace; HSV: Hue Saturation Value colorspace; HOG: Histogram of Oriented Gradients; ISM: Implicit Shape Model; WFD: Weighted Frame Difference; SIFT: Scale-Invariant Feature Transform; I: Implicit; A: Automatic; T&T: Tag and track; '-': no information provided.

Ref	Feature extraction				Motion model		Sequential	Batch	TI	TT
	BS	Color	Shape	Texture	Pre-learned	Fixed	localization	association		
[34]		RGB	Edgelets			✓	✓	✓	I	I
[91]		RGB	Edgelets			✓		✓	I	I
[48]		RGB	Edgelets			✓		✓	I	I
[43]		RGB	Edgelets+HOG	Covariance matrix		✓	✓	✓	I	I
[44]		RGB	Edgelets+HOG	Covariance matrix		✓	✓	✓	I	I
[87]		RGB	Edgelets+HOG	Covariance matrix	✓		✓	✓	I	I
[14]						✓		✓	I	I
[62]		HSV				✓		✓	I	I
[92]	Gaussian	RGB				✓	✓		A	A
[59]		HSV				✓	✓		A	-
[17]		RGB				✓	✓		A	A
[8]	WFD					✓	✓		A	A
[10]			HOG			✓	✓	✓	I	I
[67]			HOG		✓		✓		T&T	-
[66]		-			✓		✓		T&T	-
[2]		-			✓		✓		T&T	-
[41]			Gradient		✓		✓		T&T	-
[88]		RGB	Elliptical model	SIFT		✓	✓		A	-
[12]			HOG/ISM			✓	✓		A	A
[30]	-	RGB				✓	✓		A	-
[1]		HSV				✓	✓		A	A
[73]	Gaussian+Vessel					✓	✓		A	A
[31]		HSV				✓	✓		A	A

each tracklet. The M-step links tracklets using the information from the E-step and long tracks are obtained through Hungarian algorithm [34, 87, 90]. The assignment matrix used by the Hungarian algorithm is formulated as a MAP problem, relying on link probabilities calculated with associated detection responses.

1.9 Summary and outlook

In this chapter we discussed state-of-the-art multi-target trackers and presented a general flow diagram that allowed us to highlight the major tracking steps. We also presented a survey on recent multi-target trackers, discussing their major steps that include feature extraction algorithms, prediction models, localization and association methods, and techniques for track initialization and termination. We finally discussed how contextual information can be employed to improve tracking performance.

Interesting open challenges in multi-target tracking include the effective extension of feature selection for target-background separability from offline [74] to on-line approaches [70], defining motion models that are flexible to deal with different dynamics of a scene [67], and predicting tracking failures by identifying image regions where trackers are likely to fail [37]. These failures can be detected by employing interaction models based on track information [39] and potentially solved by strengthening the trackers with methods for self-tuning parameters [46] (e.g. resampling strategy for particle filter [65]). Removing the dependence of user interaction is also desirable to make the environment learning stage flexible to context changes [38, 56]

and independent from user feedbacks [51].

Finally, there is a growing interest in tracking targets using multiple cameras for increasing the overall field of view [78]. In this case, the target discrimination and identity association techniques need to consider the appearance variability of targets across cameras due to changes in illumination, pose and appearance.

Bibliography

[1] Ali, I., Dailey, M. N., Sep. 2009. Multiple human tracking in high-density crowds. In: Proc. of Conference on Advanced Concepts for Intelligent Vision Systems. Bordeaux, France,.

[2] Ali, S., Shah, M., Oct. 2008. Floor fields for tracking in high density crowd scenes. In: Proc. of European Conference on Computer Vision. Marseille, France.

[3] Andrieu, C., Freitas, N. D., Doucet, A., Jordan, M. I., 2003. An introduction to mcmc for machine learning. Machine Learning 50 (1), 5–43.

[4] Anjum, N., Cavallaro, A., Nov. 2008. Multi-feature object trajectory clustering for video analysis. IEEE Trans. on Circuits and Systems for Video Technology 18 (11), 1555–1564.

[5] APIDIS dataset, http://www.apidis.org/Dataset/, last accessed: 11 November 2012.

[6] Arulampalam, M., Maskell, S., Gordon, N., Clapp, T., Aug. 2002. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. on Signal Processing 50 (2), 174–188.

[7] Bar-Shalom, Y., 1988. Tracking and data associaion. Academic Press Professional, Inc.

[8] Beleznai, C., Fruhstuck, B., Bischof, H., Apr. 2006. Human tracking by fast mean shift mode seeking. Journal of Multimedia 1 (1), 1–8.

[9] Beleznai, C., Schreiber, D., Sept. 2010. Multiple object tracking by hierarchical association of spatio-temporal data. In: Proc. of International Conference on Image Processing. Hong Kong, China.

[10] Benfold, B., Reid, I., Jun. 2011. Stable multi-target tracking in real-time surveillance video. In: Proc. of Computer Vision and Pattern Recognition. Colorado Springs, USA.

[11] Besag, J., 1986. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society Series B Methodological 48 (3), 259–302.

[12] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L. V., Sep. 2011. Online multi-person tracking-by-detection from a single, uncalibrated camera. IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (9), 1820–1833.

[13] CAVIAR dataset, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, last accessed: 11 November 2012.

[14] Collins, R., Jun. 2012. Multitarget data association with higher-order motion models. In: Proc. of IEEE Computer Vision and Pattern Recognition. Providence, Rhode Island, USA.

[15] Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., et al., D. D., 2000. A system for video surveillance and monitoring. Tech. Rep. CMU-RI-TR-00-12, The Robotics Institute, Carnegie Mellon University, Pittsburgh PA.

[16] Comaniciu, D., Ramesh, V., Meer, P., May 2003. Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 25 (5), 564–577.

[17] Czyz, J., Ristic, B., Macqa, B., Aug. 2007. A particle filter for joint detection and tracking of color objects. Image and Vision Computing 25, 1271–1281.

[18] Dalal, N., Triggs, B., Jan. 2005. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition. San Diego, CA, USA.

[19] Daniyal, F., Cavallaro, A., Nov. 2011. Multi-camera scheduling for video production. In: Proc. of Conference on Visual Media Production. London, UK.

[20] Dollar, P., Wojek, C., Schiele, B., Perona, P., Apr. 2012. Pedestrian detection: an evaluation of the state of the art. Trans. on Pattern Analysis and Machine Intelligence 34 (4), 743–761.

[21] Doucet, A., de Freitas, J., Gordon, N., 2001. Sequential Monte Carlo methods in practice. Springer-Verlang, Ch. An introduction to sequential Monte Carlo methods.

[22] Duda, R., Hart, P., Stork, D., 2001. Pattern Classification.

[23] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., Ramanan, D., Sep. 2010. Object detection with discriminatively trained part based models. IEEE Trans. on Pattern Analysis and Machine Intelligence 32 (9), 1627–1645.

[24] Frangi, A., Niessen, W., Vinken, K., Viergever, M., Oct. 1998. Multiscale vessel enhancement filtering. In: Proc. of Medial Image Computing and Computer-Assisted Intervention. Cambridge, MA, USA.

[25] Freund, Y., Iyer, R., Schapire, R., Singer, Y., Jan. 2003. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4 (6), 933–969.

[26] Gall, J., Yao, A., Razavi, N., Gool, L. V., Lempitsky, V., Nov. 2011. Hough forests for object detection, tracking, and action recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (11), 2188–2202.

[27] Goldberg, A., Jan. 1997. An efficient implementation of a scaling minimum-cost flow algorithms. Journal of Algorithms 22 (1), 1–29.

[28] Gonzalez, R. C., Woods, R. E., 2008. Digital Image Processing. Pearson Prentice Hall.

[29] Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge university press.

[30] Hess, R., Fern, A., Jun. 2009. Discriminatively trained particle filters for complex multi-object tracking. In: Proc. of IEEE Computer Vision and Pattern Recognition. Miami, FL, USA.

[31] Hoseinnezhad, R., Vo, B.-N., Suter, D., Vo, B.-T., March 2010. Multi-object fitlering from image sequence without detection. In: Proc. of International Conference on Acoustic, Speech and Signal Processing. Dallas, Texas, USA.

[32] Hu, M.-C., Chang, M.-H., Wu, J.-L., Chi, L., Mar. 2011. Robust camera calibration and player tracking in broadcast basketball video. IEEE Trans. on Multimedia 13 (2), 266–279.

[33] Hu, W., Xiao, X., Fu, Z., Tan, D. X. T., Maybank, S., Sept. 2006. A system for learning statistical motion pattern. IEEE Trans. on Pattern Analysis and Machine Intelligence 28 (9), 1450–1464.

[34] Huang, C., Wu, B., Nevatia, R., Oct. 2008. Robust object tracking by hierarchical association of detection responses. In: Proc. of European Conference on Computer Vision. Marseille, France.

[35] Hue, C., Cadre, J.-P. L., Perez, P., Feb. 2002. Sequential monte carlo methods for multiple target tracking and data fusion. IEEE Trans. on Signal Processing 50 (2), 309–325.

[36] iLids dataset, http://www.eecs.qmul.ac.uk/ andrea/avss2007_d.html, last accessed: 11 November 2012.

[37] Kalal, Z., Mikolajczyk, K., Matas, J., Aug. 2010. Forward-backward error: Automatic detection of tracking failures. In: Proc. of International Conference on Pattern Recognition. Istambul, Turkey.

[38] Kembhavi, A., Yeh, T., Davis, L. S., Sep. 2010. Why did the people cross the road (there)? scene understanding using probabilistic logic models and common sense reasoning. In: Proc. of European Conference on Computer Vision. Crete, Greece.

[39] Khan, Z., Balch, T., Dellaert, F., Nov. 2005. MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Trans. on Pattern Analysis and Machine Intelligence 27 (11), 1805–1819.

[40] Kindermann, R., Snell, J. L., Providence, Rhode Island, 2000. Markov Random Fields and their applications. American Mathematical Society.

[41] Kratz, L., Nishino, K., May 2012. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 34 (5), 987–1002.

[42] Kuhn, H., 1955. The hungarian method for the assignement problem. Naval Research Logistics Quarterly 2, 843–854.

[43] Kuo, C., Huang, C., Nevatia, R., Jun. 2010. Multi-target tracking by on-line learned discriminative appearance models. In: Proc. of IEEE Computer Vision and Pattern Recognition. San Francisco, CA, USA.

[44] Kuo, C., Nevatia, R., 20-25 Jun. 2011. How does person identity recognition help multi-person tracking? In: Proc. of IEEE Computer Vision and Pattern Recognition. Colorado Springs, USA, 2011, pp. 1217–1224.

[45] Leibe, B., Schindler, K., Cornelis, N., Gool, L. V., Oct. 2008. Coupled object detection and tracking from static cameras and moving vehicles. IEEE Trans. on Pattern Analysis and Machine Intelligence 30 (10), 1683–1698.

[46] Li, M., Tan, T., Chen, W., Huang, K., 2012. Efficient object tracking by incremental self-tuning particle filtering on the affine groups. IEEE Trans. on Signal Processing 21, 1298–1313.

[47] Li, M., Zhang, Z., Huang, K., Tan, T., Nov. 2009. Rapid and robust human detection and tracking based on omega-shape features. In: Proc. of IEEE International Conference on Image Processing. Cairo, Egypt.

[48] Li, Y., Huang, C., Nevatia, R., Jun. 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: Proc. of Computer Vision and Pattern Recognition. Miami, FL, USA.

[49] Li, Y., Nevatia, R., Oct. 2008. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In: Proc. of European Conference on Computer Vision. Marseille, France.

[50] Lowe, D., Sept. 1999. Object recognition from local scale-inveriant feature. In: Proc. of IEEE International Conference on Computer Vision. Corfu, Greece.

[51] Maggio, E., Cavallaro, A., Aug. 2009. Learning scene context for multiple object tracking. IEEE Trans. on Image Processing 18 (8), 1873–1884.

[52] Maggio, E., Cavallaro, A., 2011. Video tracking: theory and practice. Wiley.

[53] Maggio, E., Smeraldi, F., Cavallaro, A., Oct. 2007. Adaptive multi-feature tracking in a particle filtering framework. IEEE Transactions on Circuits and Systems for Video Technology 17 (10), 1348–1359.

[54] Mahler, R., 1994. Random-set approach to data fusion. SPIE.

[55] Mahler, R., 2002. A theoretical foundation for the stein-winter probability hypothesis density (phd) multitarget tracking approach. In: Proc. of MSS National Symposium on Sensor and Data Fusion. San Antonio, Texas, USA.

[56] Makris, D., Ellis, T., Jun. 2005. Learning semantic scene models from observing activity in visual surveillance. IEEE Trans. on Systems, Man and Cybernetics - Part B 35 (3), 397408.

[57] Mazzon, R., Tahir, S. F., Cavallaro, A., 2012 (to appear). Person re-identification in crowd. Pattern Recognition Letters.

[58] Mitiche, A., Bouthemy, P., 1996. Computation and analysis of image motion: A synopsis of current problems and methods. International Journal of Computer Vision 19 (1), 29–55.

[59] Okuma, K., Talenghani, A., Freitas, N. D., May 2004. A boosed particle filter: multitarget detection and tracking. In: Proc. of European Conference on Computer Vision. Prague, Czech Republic.

[60] Papoulis, A., Pillai, S., 2002. Probability, random variables and stochastic processes. Mc Graw Hill.

[61] Piccardi, M., Oct. 2004. Background subtraction techniques: a review. In: Proc. of International Conference on Systems, Man and Cybernetics. The Hauge, Netherlands.

[62] Qin, Z., Shelton, C., Jun. 2012. Improving multi-target tracking via social grouping. In: Proc. of IEEE Computer Vision and Pattern Recognition. Provence, Rhode Island, USA.

[63] Rabiner, L., Feb. 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proceeding of the IEEE 77 (0018-9219), 257–286.

[64] Reid, D., 1979. An algorithm for tracking multiple targets. IEEE Trans. on Automatic Control 24 (6), 843–854.

[65] Ristic, B., Arulampalam, S., Gordon, N., 2004. Beyond the Kalman filter: particle filters for tracking applications. Artech House.

[66] Rodriguez, M., Ali, S., Kanade, T., Sep 2009. Tracking in unstructured crowded scenes. In: Proc. of IEEE International Conference on Computer Vision. Kyoto, Japan.

[67] Rodriguez, M., Laptev, I., Sivic, J., Audibert, J., Nov. 2011. Density-aware person detection and tracking in crowds. In: Proc. of IEEE International Conference on Computer Vision. Barcelona, Spain.

[68] Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.-Y., Nov. 2011. Data-driven crowd analysis in videos. In: Proc. of IEEE International Conference on Computer Vision. Barcelona, Spain.

[69] Sabzmeydani, P., Mori, G., 2007 2007. Detecting pedestrians by learning shapelet features. In: Proc. of IEEE Computer Vision and Pattern Recognition. Minneapolis, Minnesota, USA.

[70] Salti, S., Cavallaro, A., Stefano, L. D., (to appear). Adaptive appearance modeling for video tracking: survey and evaluation. IEEE Trans. of Image Processing.

[71] Schapire, R., Singer, Y., 1999. Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning.

[72] Smith, K., Ba, S., Odobez, J.-M., Gatica-Perez, D., Jul. 2008. Tracking the visual focus of attention for a varying number of wandering people. IEEE Trans. on Pattern Analysis and Machine Intelligence 30 (7), 1–17.

[73] Stalder, S., Grabner, H., Gool, L. V., Sep. 2010. Cascaded confidence filtering for improved tracking-by-detection. In: Proc. of European Conference on Computer Vision. Crete, Greece.

[74] Stenger, B., Woodley, T., Cipolla, R., Jun. 2009. Learning to track with multiple observers. In: Proc. of IEEE Computer Vision and Pattern Recognition. Washington, DC, USA.

[75] Suk, H.-I., Jain, A., Lee, S.-W., 2011. A network of dynamic probabilistic models for human interaction analysis. IEEE Trans. on Circuits and Systems for Video Technology 21, 932–945.

[76] Taj, M., Cavallaro, A., Sept. 2009. Multi-camera track-before-detect. In: Proc. of Conference on Distributed Smart Cameras. Como, Italy.

[77] Taj, M., Cavallaro, A., 2010. Recognizing interactions in video. Vol. 282/2010. Intelligent Multimedia Analysis for Security Applications, Springer.

[78] Taj, M., Cavallaro, A., May 2011. Distributed and decentralized multi-camera tracking. IEEE Signal Processing Magazine 28 (3), 46–58.

[79] Tomasi, C., Kanade, T., Apr. 1991. Detection and tracking of point features. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University.

[80] Tuzel, O., Porikli, F., Meer, P., Oct. 2008. Pedestrian detection via classification on Riemannian manifolds. IEEE Trans. on Pattern Analysis and Machine Intelligence 30 (10), 1–15.

[81] Vermaak, J., Doucet, A., Perez, P., Oct. 2003. Maintaining multi-modality through mixture tracking. In: Proc. of IEEE International Conference on Computer Vision. Nice, France.

[82] Viola, P., Jones, M., Snow, D., 2005. Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision 63 (2), 153–161.

[83] Vo, B.-N., Vo, B.-T., Pham, N.-T., Suter, D., Oct. 2010. Joint detection and estimation of multiple objects from image observations. IEEE Trans. on Signal Processing 58 (10), 5129–5241.

[84] Waibel, A., Schultz, T., Bett, M., Denecke, M., Malkin, R., Rogina, I., Stiefelhagen, R., Apr. 2003. Smart: the smart meeting room task at isl. In: Proc. of Conference on Acoustics, Speech, and Signal Processing. Hong Kong, China.

[85] Wang, X., Ma, X., Grimson, W., Mar. 2009. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian model. IEEE Trans. of Pattern Analysis and Machine Intelligence 31 (3), 539–555.

[86] Wu, B., Nevatia, R., Nov. 2007. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. Internation Journal of Computer Vision 75 (2), 247–266.

[87] Yang, B., Nevatia, R., 16-21 Jun. 2012. Multi-target tracking by online learning of non-linear motion patterns and robust appearance model. In: Proc. of IEEE Computer Vision and Pattern Recognition. Providence, Rhode Island, USA, 2012.

[88] Yang, M., Lv, F., Xu, W., Gong, Y., Sep. 2009. Detection driven adaptive multi-cue integration for multiple human tracking. In: Proc. of IEEE International Conference on Computer Vision. Kyoto, Japan.

[89] Yilmaz, A., Javed, O., Shah, M., Dec. 2006. Object tracking: a survey. Journal ACM Computing Surveys 38 (4), 1–45.

[90] Yu, Q., Medioni, G., 2009. Motion pattern interpretation and detection for tracking moving vehicles in airborne videos. In: Proc. of Computer Vision and Pattern Recognition.

[91] Zhang, L., Li, Y., Nevatia, R., Jun. 2008. Global data association for multi-object tracking using network flows. In: Proc. of IEEE Computer Vision and Pattern Recognition. Anchorage, Alaska, USA.

[92] Zhao, T., Nevatia, R., Jul. 2004. Tracking multiple humans in crowded environments. In: Proc. of Computer Vision and Pattern Recognition. Washington, DC, USA.

[93] Zhou, B., Wang, X., Tang, X., Jun. 2011. Random field topic model for semantic region analysis in crowded scenes from tracklets. In: Proc. of IEEE Computer Vision and Pattern Recognition. Colorado Springs, USA.

[94] Zhu, Q., Avidan, S., Yeh, M.-C., Cheng, K.-T., Jun. 2006. Fast human detection using a cascade of histograms of oriented gradients. In: Proc. of IEEE Computer Vision and Pattern Recognition. New York, USA.