Robust Wide Area Tracking in Single and Multiple Views

B. Song, R. J. Sethi and A. K. Roy-Chowdhury

Abstract Maintaining the stability of tracks on multiple targets in video over extended time periods and wide areas remains a challenging problem. Basic trackers like the Kalman filter or particle filter deteriorate in performance as the complexity of the scene increases. A few methods have recently shown encouraging results in these application domains. They rely on learning context models, the availability of training data, or modeling the inter-relationships between the tracks. In this chapter, we provide an overview of research in the area of long-term tracking in video. We review some of the methods in the literature and analyze the common sources of errors which cause trackers to fail. We also discuss the limits of performance of the trackers as multiple objects come together to form groups and crowds. On multiple real-life video sequences obtained for a single camera as well as a camera network, we compare the performance of some of the methods.

1 Introduction

Tracking can be defined as a problem of locating a moving object (or multiple objects) over time in the image plane. In other words, the objective of a tracker is to associate target objects in consecutive video frames so as to determine the identities and locations of objects in the video sequence. Multiple object tracking is the most fundamental task for higher level automated video content analysis for its wide application in human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, and video editing.

Bi Song
University of California, Riverside CA 92521 e-mail: bsong@ee.ucr.edu

Ricky J. Sethi
University of California, Los Angeles CA 90095 e-mail: rickys@sethi.org

Amit K. Roy-Chowdhury
University of California, Riverside CA 92521 e-mail: amitrc@ee.ucr.edu

Some of the most basic tracking methods include the Kalman filter, particle filter and mean shift tracker. However, by themselves, these methods are usually not able to track over extended space-time horizons.

In addition to challenges in tracking a single object, like occlusion, appearance variation, and image noise, the critical issue in multi-target tracking is data association, i.e., the problem of linking a sequence of object observations together across image frames. Although a large number of trackers exist, their reliability falls off quickly with the length of the tracks. Stable, long-term tracking is still a challenging problem. Moreover, for multiple targets, we have to consider the interaction between the targets which may cause errors like switching between tracks, missed detections and false detections. In addition, wide area tracking over a camera network introduces certain challenges that are unique to this particular application scenario, like handoff between cameras; often, errors are caused in this handoff stage. Therefore, detection and correction of the errors in the tracks is the key to robust long term and wide area tracking.

In this chapter, we start off with a review of current work in multi-target tracking. We briefly describe two most basic stochastic tracking methods – the Kalman Filter and Particle Filter, as well as two representative data association methods – Multi-Hypothesis Tracking (MHT) and Joint Probabilistic Data Association Filters (JPDAF) methods. We then analyze two common sources of errors, which allow us to identify tracklets (i.e., the short-term fragments with low probability of error). The long-term tracking problem can now be defined as developing approaches on how to associate the traklets based on their features. Similar ideas can be applied to camera networks as tracking across non-overlapping camera networks is essentially to find the association of the targets observed in different camera views, i.e., the handoff problem. We briefly describe a recent method that provides an optimization framework and strategy for computing the associations between tracklets as a stochastic graph evolution scheme. This can be used to obtain tracks of objects that have been occluded or are difficult to disambiguate due to clutter or appearance variations. For a non-overlapping camera network, by associating the tracks from different cameras using this stochastic graph evolution framework, it will automatically lead to a solution of the handoff problem. Finally, we provide a numerical comparison of some of the approaches.

The remainder of this chapter is organized as follows: A review of tracking in single cameras is provided in Sec. 2. Then in Sec. 3, we analyze the common sources of errors in tracking. We review the issues in tracking across a camera network in Sec. 4. In Sec. 5, we briefly describe a novel tracklet association strategy using stochastic graph evolution. Sec. 6 shows how to identify groups and crowds where the tracking method may perform poorly, which will allow a tracker to automatically switch to a different strategy. We show comparison of different tracking methods and experimental results on tracking in a camera network in Sec. 7. We conclude our work in Sec. 8 with a description of future work.

2 Review of Multi-Target Tracking Approaches

In this section, we review the literature on multi-target tracking. We start by describing two of the most basic methods - the Kalman Filter [19] and Particle filter [15]. They are stochastic methods and solve tracking problems by taking the measurement and model uncertainties into account during object state estimation. They have been extensively used in the vision community for tracking, but these methods are not powerful for tracking multiple objects by themselves, e.g., the Kalman filter and particle filter assume a single measurement at each time instant. In [14], particle filters were used to track multiple objects by incorporating probabilistic MHT [31] for data association. We describe the MHT [31] and JPDAF[4] strategies for tracking multiple targets.

2.1 Kalman Filter Based Tracker

Consider a linear dynamical system with the following time propagation and observation models for a moving object in the scene:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t); \quad \mathbf{x}(0) \tag{1}$$

$$\mathbf{z}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{v}(t) \tag{2}$$

where $\mathbf{x}$ is the state of the target, $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are zero mean white Gaussian noise ($\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}^l), \mathbf{v}(t) \sim \mathcal{N}(0, \mathbf{R})$) and $\mathbf{x}(0) \sim \mathcal{N}(\mathbf{x}_0, \mathbf{P}_0)$ is the initial state of the target.

Kalman filtering is composed of two steps, prediction and correction. The prediction step uses the state model to predict the new state of the variables:

$$\begin{aligned} \bar{\mathbf{P}}(t+1) &= \mathbf{A}\mathbf{P}(t)\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T, \\ \bar{\mathbf{x}}(t+1) &= \mathbf{A}\hat{\mathbf{x}}(t). \end{aligned} \tag{3}$$

The correction step uses the current observations $\mathbf{z}(t)$ to update the objects state:

$$\begin{aligned} \mathbf{K}(t+1) &= \bar{\mathbf{P}}(t+1)\mathbf{F}^T(\mathbf{F}\bar{\mathbf{P}}(t+1)\mathbf{F}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}(t+1) &= \bar{\mathbf{x}}(t+1) + \mathbf{K}(t+1)(\mathbf{z}(t+1) - \mathbf{F}\bar{\mathbf{x}}(t+1)), \\ \hat{\mathbf{P}}(t+1) &= (\mathbf{I} - \mathbf{K}(t+1)\mathbf{F})\bar{\mathbf{P}}(t+1). \end{aligned} \tag{4}$$

Extensions to this basic approach dealing with non-linear models in video applications can be found in [41].

2.2 Particle Filter Based Tracker

The particle filter is often used in tracking applications in video to deal with non-linear and/or non-Gaussian models [2]. The following are the main steps typically used in a particle filter based tracker with some variations.

Moving objects are often initialized using motion detection. The background modeling algorithm in [40] can be used for its adaptability to illumination change, and to learn the multimodal background through time. In addition, in many applications, by observing that most of the interested targets, like people and vehicles, are on ground plane, the rough ground plane area can be estimated [11]. Based on the ground plane information, false alarms can be removed significantly. The target regions can then be represented by rectangles with the state vector $X_t = [x, y, \dot{x}, \dot{y}, l_x, l_y]$, where (x, y) and $(\dot{x}, \dot{y})$ are the position and velocity of a target in the x and y directions respectively, and (l_x, l_y) denote the size of the rectangle.

The observation process is defined by the likelihood distribution, $p(I_t|X_t)$, where X_t is the state vector and I_t is the image observation at t. The observation models can be generated in many ways. Here we provide an example by combining an appearance and a foreground response model, i.e.,

$$p(I_t|X_t) = p(I_t^a, I_t^f|X_t), \tag{5}$$

where I_t^a is the appearance information of I_t and I_t^f is the foreground response of I_t using a learned background model. I_t^f is a binary image with "1" for foreground and "0" for background. It is reasonable to assume that I_t^a and I_t^f are independent and thus (5) becomes

$$p(I_t|X_t) = p(I_t^a|X_t)p(I_t^f|X_t).$$

The appearance observation likelihood can be defined as

$$p(I_t^a|X_t) \propto \exp\{-B(ch(X_t), ch_0)^2\},$$

where $ch(X_t)$ is the color histogram associated with the rectangle region of X_t and ch_0 is color histogram of the initialized target. $B(.)$ is the Bhattachayya distance between two color histograms. The foreground response observation likelihood can be defined as

$$p(I_t^f|X_t) \propto \exp\{-(1 - \frac{\#F(X_t)}{\#X_t})^2\},$$

where $\#F(X_t)$ is the number of foreground pixels in the rectangular region of X_t and $\#X_t$ is the total number of pixels in that rectangle. $\frac{\#F(X_t)}{\#X_t}$ represents the percentage of the foreground in that rectangle. The observation likelihood would be higher if more pixels in the rectangular region of X_t belong to the foreground. The reader should note that these are representative examples only. Various models are possible, and indeed, have been used in the literature.

The particle filter (PF) is a sequential Monte Carlo method (sequential importance sampling plus resampling) which provides at each t, an N sample Monte Carlo

approximation to the prediction distribution, $\pi_{t|t-1}(dx) = Pr(X_t \in dx|I_{1:t-1})$, which is used to search for newly observed targets. These are then used to update $\pi_{t|t-1}$ to get the filtering (posterior) distribution, $\pi_{t|t}(dx) = Pr(X_t \in dx|I_{1:t})$. A particle filter is used because the system and observation models are nonlinear and the posterior can temporarily become multi-model due to background clutter.

2.3 Multi-Hypothesis Tracking (MHT)

This algorithm allows multiple hypotheses to be propagated in time as data is received. MHT is an iterative algorithm and is initialized with a set of current track hypotheses. Each hypothesis is a collection of disjoint tracks. For each hypothesis, the position of each object at the next time step is predicted. On receiving new data, each hypothesis is expanded into a set of new hypotheses by considering all measurement-to-track assignments for the tracks within the hypothesis. The probability of each new hypothesis is calculated. Often, for reasons of finite computer memory and computational power, the most unlikely hypotheses are deleted. The final tracks of the objects are the most likely set of associations over the time period. Note that MHT exhaustively enumerates all possible associations and is computationally exponential both in memory and time.

2.4 Joint Probabilistic Data Association Filters (JPDAF)

This method is specifically designed for cluttered measurement models. The idea is to compute an expected state estimate over the various possibilities of measurement-to-track associations. Assuming we have n tracks and m measurements at time t, $Z(t) = \{z_1(t), \ldots, z_m(t)\}$, the state estimation of target i is

$$\hat{x}_i(t) = E[x_i(t)|Z(t)] = \sum_{j=1}^{m} E[x_i(t)|\chi_{ij}^t, Z(t)]P(\chi_{ij}^t|Z(t))$$

where χ_{ij} denotes the event that measurement i associates to target j.

In order to overcome the large computational cost of MHT and JPDAF, various optimization algorithms such as Linear Programming [17], Quadratic Boolean Programming [23], and Hungarian algorithm [27] are used for data association. In [45], data association was achieved through a MCMC sampling based framework. In [34], a multiframe approach was proposed to preserve temporal coherency of the speed and position. They formulate the correspondence as a graph theoretic problem to finding the best path for each point across multiple frames. They use a window of frames during point correspondence to handle occlusions whose durations are shorter than the temporal window used to perform matching.

3 Errors in Multi-Target Tracking

Tracking is a state estimation problem and errors are inevitable in even carefully designed strategies. Therefore, it is important to understand the sources of the errors and mitigate their effects as far as possible. There are two common errors: lost track (when the track is no longer on any target, but on the background) and track switching (when targets are close and the tracks are on the wrong target); this includes tracks merging and splitting. Identifying these situations can lead to the rules for tracklet estimation, i.e., determining short track segments where the probability of an error is low. An example is shown in Fig. 1. We describe these two common sources of errors.

Detection of lost track: The tracking error (TE) [4] or prediction error is the distance between the current observation and its prediction based on past observations. TE will increase when the tracker loses track and can be used to detect the unreliability of the track result. As an example, in the preceding observation model for the particle filter, TE of tracked target $\hat{X}_t$ is calculated by

$$TE(\hat{X}_t, I_t) = TE_a(\hat{X}_t, I_t) + TE_f(\hat{X}_t, I_t), \tag{6}$$

$$\text{where } TE_a(\hat{X}_t, I_t) = B(ch(X_t), ch_0)^2 \text{ and } TE_f(\hat{X}_t, I_t) = \left(1 - \frac{\#F(X_t)}{\#X_t}\right)^2.$$

If a lost track is detected, it means the tracking result after this point is not reliable; in the tracking procedure, we can stop doing tracking after this point and identify a tracklet. In the case of false detection (i.e., the detected target is a part of background), or target passes through a region with similar color, or a target stops, the background modeling algorithm will adapt to treat this as a part of the background, and thus TE_f will eventually increase. Then a lost track will be detected.

Track Switching: When targets are close to each other, a track switch can happen with high probability especially if the appearances of targets are similar. Thus, we can inspect the distances between targets, and break the tracks into tracklets at the points where targets are getting close, as shown in Fig. 1.

3.1 Solution Strategies

Many state-of-the-art tracking algorithms focus on how to avoid errors. In [46], the authors proposed a min-cost flow framework for global optimal data association. A tracklet association based tracking method was presented in [8], which fixed the affinity model heuristically and focused on searching for optimal associations. A HybridBoosted affinity model was learned in [25]. The method is built on the availability of training data under a similar environment, which may not be always feasible. The authors in [3] addressed the problem of learning an adaptive appearance model for object tracking. Context information was considered in [44] to help

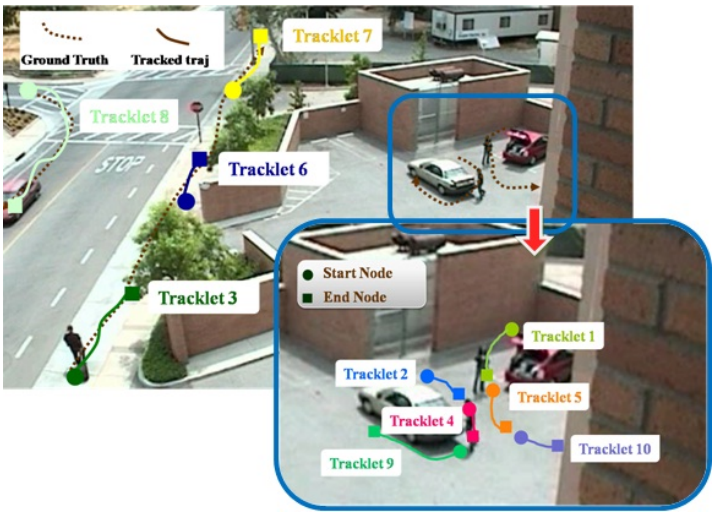


Fig. 1 An example of tracklet identification. The ground truth trajectories are represented by brown dotted lines. The estimated tracklets due to detection of a lost track (track of the person in lower left corner due to occlusion) and targets' close proximity (the persons moving around the cars) are clearly shown in different colors.

in tracking, by integrating a set of auxiliary objects which are learned online. It can be a powerful method in applictions where it is easy to find these auxiliary objects. A joint probabilistic relation graph approach was presented in [42] to simultaneously detect and track a large number of vehicles in low frame rate aerial videos. The authors explored vehicle behavior model from road structure and generated a set of constraints to regulate both object based vertex matching and pairwise edge matching schemes. These two matching schemes were unified into a single cost minimization framework to produce a quadratic optimized association result. In [5], an inference graph was built to represent merge-split relations between the tracked blobs, so as to handle fragmentation and grouping ambiguity in tracking. In [35], an adaptive tracklet association was proposed and is explained later in 5.

3.2 Tracklet Affinity Modeling

Computing long-term associations between tracklets requires evaluating similarities between them. The estimated tracklets need to be associated based on their affinities to come up with longer tracks. The affinities between tracklets are often modeled by exploring their appearance and motion attributes. Here we provide some examples for achieving this.

Appearance model: The appearance affinity between a pair of tracklets, (T_i, T_j), can be defined based on their color histograms. Let Ch_i and Ch_j be the mean color

histograms learned within T_i and T_j. Then the appearance affinity can be defined as

$$A_a(T_i, T_j) \propto \exp(-B(Ch_i, Ch_j)), \tag{7}$$

where $B(.)$ represents the Bhattacharya distance between two histograms.

Motion model: As described in [43], the motion affinity can be modeled based on both the forward and backward velocities of the tracklets. The forward and backward velocities are estimated within each tracklet. Assume tracklet T_i occurs earlier in time, and tracklet T_i begins after the T_i ended, the motion affinity between T_i and T_j is defined as

$$A_m(T_i, T_j) \propto \exp(-(p_i^{tail} + v_i^F \Delta t - p_j^{head})^2) \exp(-(p_j^{head} + v_j^B \Delta t - p_i^{tail})^2), \tag{8}$$

where p_i^{tail} and p_j^{head} are the tail and head positions of T_i and T_j, v_i^F and v_j^B are their forward and backward velocities, and Δt is the time gap between T_i and T_j.

Since it is reasonable to assume that the appearance and motion are independent, the affinity of a pair of tracklets can be modeled as

$$A(T_i, T_j) = A_a(T_i, T_j) \cdot A_m(T_i, T_j). \tag{9}$$

4 Tracking in Camera Networks

Some of the existing methods on tracking in a camera network include [13, 16, 21]. The authors in [30] used location and velocity of objects moving across multiple non-overlapping cameras to estimate the calibration parameters of the cameras and the target's trajectory. In [24], a particle filter was used to switch between track prediction between non-overlapping cameras and tracking within a camera. In [20], the authors presented a method for tracking in overlapping stationary and pan-tilt-zoom cameras by maximizing a joint motion and appearance probability model. A Bayesian formulation of the problem of reconstructing the path of objects across multiple non-overlapping cameras was presented in [21] using color histograms for object appearance. A graph-theoretic framework for addressing the problem of tracking in a network of cameras was presented in [16]. An on-line learned discriminative appearance affinity model by adopting Multiple Instance Learning boosting algorithm was proposed in [22] for associating multi-target tracks across multiple non-overlapping cameras.

A related work that deals with tracking targets in a camera network with PTZ cameras is [29]. Here, the authors proposed a mixture between a distributed and a centralized scheme using both static and PTZ cameras in a virtual camera network environment. A framework for distributed tracking and control in camera network using Kalman-consensus filter was presented in [39, 36].

Tracking in camera networks is closely related to person re-identification in camera networks. In [10], a machine learning algorithm was used to find the best feature representation of objects, where many different kinds of simple features to be

combined into a single similarity function for matching objects. In [28], the person re-identify across disjoint camera views was reformulated as a ranking problem. By learning a subspace where the potential true match is given highest ranking rather than any direct distance measure, the problem was solved using a Ensemble RankSVM. A Cross Canonical Correlation Analysis framework was formulated in [26] to detect and quantify temporal and causal relationships between regional activities within and across camera views. In [7], the authors presented an appearance-based method for person re-identification. It consists in the extraction and fusion of features that model three complementary aspects of the human appearance: the overall chromatic content, the spatial arrangement of colors into stable regions, and the presence of recurrent local motifs with high entropy. A spatiotemporal segmentation algorithm was employed in [9] to generate salient edgels and the invariant signatures were generated by combining normalized color and salient edgel histograms for establishing the correspondence across camera views. In [37, 38], it was shown that tracking in a camera network can be solved using a stochastic adaptive strategy. Adapting the feature correspondence computations by modeling the long-term dependencies between them and then obtaining the statistically optimal paths for each person differentiates this approach from existing ones. It provides a solution that is robust to errors in feature extraction, correspondence and environmental conditions.

The main new (compared to single camera tracking) challenge in the problem of tracking across non-overlapping camera networks is to find the correspondences between the targets observed in different camera views. This is often referred to as the handoff problem in camera networks. Thus, we can think of tracking over a camera network as being equivalent to finding the associations between the tracklets obtained in different single cameras. Then, the problem boils down to finding the affinities between the tracklets so as to have tracks across cameras. Depending on the applications, various features can be used like appearance, motion, calibration, travel time, 3D models, etc. As an example, we show how the travel time between entry/exit nodes of different cameras can be used in the affinity modeling. The affinity between two tracks T_i^m and T_j^n that are observed at camera C_m and C_n respectively, can be estimated as the product of the similarity in appearance features and the travel time based similarity value, i.e.,

$$A(T_i^m, T_j^n) = A_a(T_i^m, T_j^n) A_\tau(\tau_{T_i^m, T_j^n}), \tag{10}$$

where $A_a(.)$ is the appearance affinity model as in (7) and $A_\tau(.)$ represents the transition pattern between two camera nodes [16].

5 A Stochastic Graph Evolution Framework for Tracklet Association

In this section, we show how the affinity models between tracklets (for single or multiple cameras) can be used to find associations between them so as to obtain

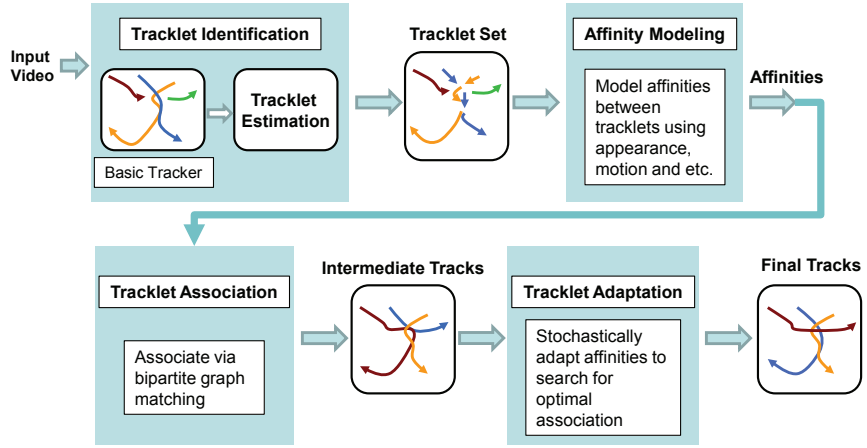
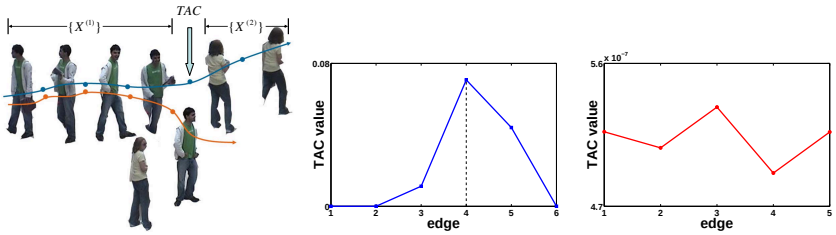


Fig. 2 Overview of stochastic graph evolution framework.

stable, robust long-term tracks. This is a brief review of a recent approach that is described in more detail in [35]. The method can deal with some of the most pressing challenges in multi-target tracking, e.g., occlusion within a view and hand-off between cameras.

Fig. 2 shows an overview of the scheme. The method begins by identifying tracklets, i.e., the short-term fragments with low probability of error, which are estimated from the initial tracks by evaluating the tracking performance as described in Section 3. The tracklets are then associated based on their affinities. Using the affinity model, a tracklet association graph (TAG) is created with the tracklets as nodes and affinity scores as weights. The association of the tracklets can be found by computing the optimal paths in the graph. The optimal path computation is based on the principles of dynamic programming and gives the maximum a posteriori (MAP) estimate of tracklets' connections as the long-term tracks for each target.

The tracking problem could be solved optimally by the above tracklet association method if the affinity scores were known exactly and assumed to be independent. However, this can be a big assumption due to well known low-level image processing challenges, like poor lighting conditions or unexpected motion of the targets. As shown in Fig. 3, if the similarity estimation is incorrect for one pair of tracklets, the overall inferred long track may be wrong even if all the other tracklets are connected correctly. This leads to a graph evolution scheme. The affinities (i.e., the weights on the edges of TAG) are stochastically adapted by considering the distribution of the features along possible paths in the association graph to search for the global optimum. A Tracklet Association Cost (TAC) function and an efficient optimization strategy are designed for this process. As shown in Fig. 3, the TAC values can be used to indicate the incorrect associations. The overall approach is able to track stably over minutes of video in challenging domains with no learning and context information.



(a) (b) (c)

Fig. 3 (a) Tracklets of two targets obtained from Videoweb courtyard dataset of Section **??**: ground truth track of the person in green T-shirt is shown with orange line, and the association results before adaptation are shown with blue line. (b)-(c): TAC values along the incorrect and correct association results respectively, (note that the range of the y-axis in (c) is much smaller than (b)). It is clear that TAC has a peak at the wrong link; thus the variance of TAC along the wrongly associated tracklets is higher than the correct one.

6 Identifying Transitions to Groups and Crowds

When a large number of targets are in close proximity, tracking each individual target becomes very difficult and sometimes it is more desirable to treat them as a group or a crowd, which can then be tracked as a single entity. As techniques designed for non-crowded scenes usually cannot be straightforwardly extended for dealing with crowded situations, crowd analysis has attracted more and more interest in recent years. A survey on crowd analysis using computer vision techniques can be found in [18]. There are also a number of research papers on tracking in crowded scenes. In [1], an approach for people tracking in structured high-density scenarios was proposed. In their work, each frame of a video sequence is divided into cells, each cell presenting one particle. A person consists of a set of particles, and each person is affected by the layout of the scene as well as the motion of other people. The method described in [12] detects global motion patterns by constructing super tracks using flow vectors for tracking high-density crowd flows in low-resolution videos. A tracking method to deal with unstructured environments was proposed in [32], in which the motion of a crowd appears to be random with different participants moving in different directions over time (e.g., a crossway). They employ the correlated topic model (CTM), which allows each location of the scene to have various crowd behaviors.

The methods on crowd analysis assume that it is known that the scene consists of a crowd. Often, we have situations where individuals merge together form a group and cannot be tracked separately any more. This detection of transitions from individuals to groups and crowds has received lesser attention. One such transitions are identified, pre-existing group/crowd analysis approaches, such as [12, 1, 32], can be employed to examine the group/crowd's dynamics.

An idea that is being currently being explored by the authors [33] involves a physics-inspired methodology to model the transition of Individuals to Groups to Crowds analogous to the transition of Individual Particles to N-Body to Fluids in

fluid dynamics, as shown in Figure 4, where the Group Transition Ratio (G_{tr}) categorizes the collection as individual people or groups. G_{tr} is defined based on comparing the distance between objects and their sizes. It utilizes the ideas in fluid dynamics for analysis of multi-object activities and, in a similar fashion, when $G_{tr} \ll 1$ for a collection of objects, we label them as a crowd; when $G_{tr} \gg 1$, we label them as individual objects; finally, when $G_{tr} \sim 1$ (empirically between 0.1 and 10), the objects are identified to be in a transition region and labeled as a group.

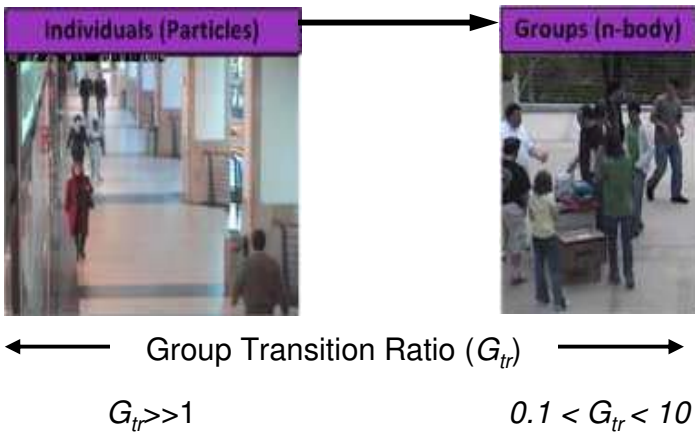


Fig. 4 Physics-inspired model of transitions from individuals to groups: we model individuals as free particles and groups as an n-body. In these two examples, the average G_{tr} for the sequence of individuals is 23.83 while the average G_{tr} for the sequence of groups is 0.18.

7 Performance Analysis

In this section, we provide comparison of some methods on multi-target tracking in single camera view and show tracking results in a camera network using the stochastic data association strategy in [37, 38].

7.1 Single Camera Tracking Performance

We compare the performance of several methods on the CAVIAR dataset. The CAVIAR (http:// homepages.inf.ed.ac.uk/rbf/CAVIARDATA1) is captured in a shopping mall corridor with heavy inter-object occlusion. To evaluate the performance quantitatively, we adopt the evaluation metrics for tracking defined in [25] and [46].

Table 1 Evaluation metrics

Name	Definition
GT	Num of ground truth trajectories
MT%	Mostly tracked: Percentage of GT trajectories which are covered by tracker output more than 80% in time
ML%	Mostly lost: Percentage of GT trajectories which are covered by tracker output less than 20% in time
FG	Fragments: The total Num of times that the ID of a target changed along a GT trajectory
IDS	ID switches: The total Num of times that a tracked target changes its ID with another target
RS%	Recover from short term occlusion
RL%	Recover from long term occlusion

In addition, we define RS and RL to evaluate the ability of recovering from occlusion (see Table 1).

In CAVIAR dataset, the inter-object occlusion is high and includes long term partial occlusion and full occlusion. Moreover, frequent interactions between targets such as multiple people talking and walking in a group make tracking more challenging. The results of [35] are shown on the relatively more challenging part of the dataset which contains 7 videos (TwoEnterShop3, TwoEnterShop2, ThreePastShop2, ThreePastShop1, TwoEnterShop1, OneShopOneWait1, OneStopMoveEnter1). Table 2 shows the comparison among the stochastic graph evolution framework [35], the min-cost flow approach in [46], HybridBoosted affinity modeling approach in [25] and a basic particle filter. It should also be noted that [25, 46] are built on the availability of training data under similar environments (e.g. 6 sequences in CAVIAR are used for training in [46]), while the stochastic graph evolution method [35] does not rely on any training. Some sample frames with results from [35] are shown in Fig. 5.

Table 2 Tracking Results on CAVIAR data set. Results of [25] and [46] are reported on 20 sequences; basic particle filter and [35] are reported on 7 most challenging sequences of the dataset. Test data used in [35] and basic particle filter has totally 12308 frames for about 500 sec.

	GT	MT	ML	FG	IDS	RS	RL
Zhang *et al.*[46]	140	85.7%	3.6%	20	15	-	-
Li *et al.*[25]	143	84.6%	1.4%	17	11	-	-
Basic particle filter	75	53.3%	10.7%	15	19	18/42	0/8
Song *et al.*[35]	75	84.0%	4.0%	6	8	36/42	6/8

7.2 Camera Network Tracking

We now show some results on tracking in a camera network using the stochastic graph evolution framework as described in detail in [38]. The network consists of 7 cameras and 26 entry/exist nodes. The cameras are installed in both indoor and

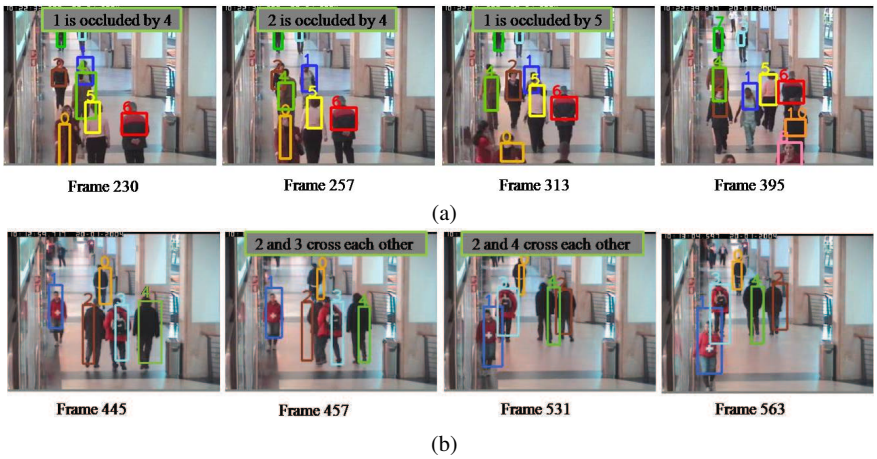


Fig. 5 Sample tracking results on CAVIAR dataset using [35].

outdoor environments which consist of large illumination and appearance changes. We considered 9 people moving across the network for about 7 minutes. Examples of some of the images of the people and entry/exit nodes are shown in Fig. 6. Note the significant changes of appearance. We also show some example tracking results on 3 targets in different colors, thus demonstrating the ability to track with handoffs. Tracking in large camera networks is still in the process of maturing and there do not exist standard datasets to compare performance of various methods. The recently released camera network dataset [6] can provide such an evaluation benchmark in the future.

8 Conclusions and Thoughts on Future Work

In this chapter, we reviewed existing methods in robust long-term tracking in single and multiple views, identified their strengthes and weaknesses, analyzed the sources of errors and discussed solution strategies. We also looked at the issue of transitions between individual tracking and group tracking. Performance comparison was provided on the well-known CAVIAR dataset.

Although tracking is one of the most studied problems in computer vision, there is still some way to go before the methods are able to work in real-world situations that involve large numbers of objects in close proximity or maintain the tracks over extended space-time horizons. Technically, this requires the methods to be robust to data association errors in cluttered scenarios, when there are large variations in appearance, or when objects are not visible due to occlusions. A promising approach is to consider contextual information, i.e., not only to look at the track of each individual object but also the collection of other nearby objects. This needs to

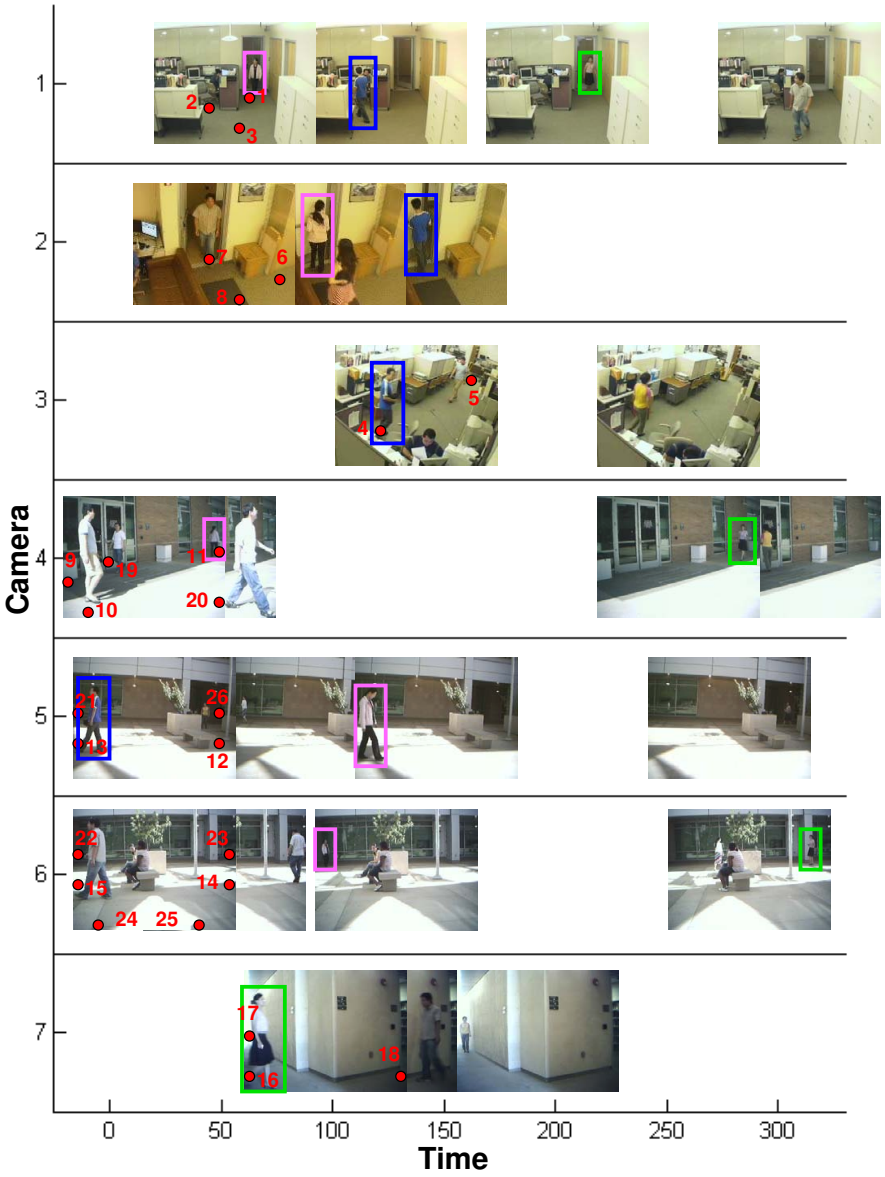


Fig. 6 Example of some of the images of the people in the network. The horizontal axis is the time when these images were observed, while the vertical axis is the index of the cameras. Some of the entry/exit nodes are marked on the images. The tracks of 3 of the targets are shown in different colors and clearly demonstrate the ability to deal with hand off in non-overlapping cameras.

be done carefully since too much emphasis on context can be misleading. Tracking

in non-overlapping camera networks has received much less attention and should be a focus area in future work. The methods should be able to scale over a large number of cameras. Standard datasets for evaluating camera network tracking need to be adopted by the tracking community. Another interesting area that has recently received interest is the development of distributed tracking frameworks, i.e., camera network tracking methods where processing is distributed over the sensor nodes [36].

9 Acknowledgements

This work was supported in part by NSF grant IIS-0712253 and subcontract from Mayachitra Inc., through a DARPA STTR award (#W31P4Q-08-C-0464).

References

1. S. Ali and M. Shah. Floor Fields for Tracking in High Density Crowd Scenes. In *Euro. Conference on Computer Vision*, 2008.
2. M. S. Arulampalam, S. Maskell, and N. Gordon. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. on Signal Processing*, 50:174–188, 2002.
3. B. Babenko, M. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
4. Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
5. B. Bose, X. Wang, and E. Grimson. Multi-Class Object Tracking Algorithm that Handles Fragmentation and Grouping. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
6. G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, , and B. Varda. VideoWeb Dataset for Multi-camera Activities and Non-verbal Communication. In *Distributed Video Sensor Networks*. Springer, 2011.
7. M. Farenzena, L. Bazzani, A. Perina, M. Cristani, and V. Murino. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.
8. W. Ge and R. Collins. Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In *British Machine Vision Conference*, 2008.
9. N. Gheissari, T. Sebastian, and R. Hartley. Person Re-Identification using Spatiotemporal Appearance. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
10. D. Gray and H. Tao. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In *Euro. Conference on Computer Vision*, 2008.
11. D. Hoiem, A. Efros, and M. Hebert. Geometric Context from a Single Image. In *IEEE Intl. Conf. on Computer Vision*, 2005.
12. M. Hu, S. Ali, and M. Shah. Detecting Global Motion Patterns in Complex Videos. In *Intl. Conf. on Pattern Recognition*, 2008.
13. T. Huang and S. Russel. Object Identification In A Bayesian Context. In *Proceeding of IJCAI*, 1997.
14. C. Hue, J. L. Cadre, and P. Prez. Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion. *IEEE Trans. on Signal Processing*, 2002.

15. M. Isard and A. Blake. Condensation - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 1998.
16. O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking Across Multiple Cameras With Disjoint Views. In *IEEE Intl. Conf. on Computer Vision*, 2003.
17. H. Jiang, S. Fels, and J. Little. A Linear Programming Approach for Multiple Object Tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
18. J.C.S. Jacques Junior, S. R. Musse, and C. R. Jung. Crowd Analysis Using Computer Vision Techniques. *IEEE Signal Processing Magazine*, September 2010.
19. R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME - Journal of Basic Engineering*, 1990.
20. J. Kang, I. Cohen, and G. Medioni. Continuous Tracking Within and Across Camera Streams. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
21. V. Kettnaker and R. Zabih. Bayesian Multi-Camera Surveillance. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999.
22. C.-H. Kuo, C. Huang, and R. Nevatia. Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models. In *Euro. Conference on Computer Vision*, 2010.
23. B. Leibe, K. Schindler, and L. V. Gool. Coupled Detection and Trajectory Estimation for Multi-Object Tracking. In *IEEE Intl. Conf. on Computer Vision*, 2007.
24. W. Leoputra, T. Tan, and F. L. Lim. Non-Overlapping Distributed Tracking using Particle Filter. In *Intl. Conf. on Pattern Recognition*, 2006.
25. Y. Li, C. Huang, and R. Nevatia. Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
26. C. Loy, T. Xiang, and S. Gong. Multi-Camera Activity Correlation Analysis. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
27. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
28. B. Prosser, W. Zheng, S. Gong, and T. Xiang. Person Re-Identification by Support Vector Ranking. In *British Machine Vision Conference*, 2010.
29. F.Z. Qureshi and D. Terzopoulos. Surveillance in Virtual Reality: System Design and Multi-Camera Control. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
30. A. Rahimi and T. Darrell. Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
31. D. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automatic Control*, 24(6):843–854, 1979.
32. M. Rodriguez, S. Ali, and T. Kanade. Tracking in Unstructured Crowded Scenes. In *IEEE Intl. Conf. on Computer Vision*, 2009.
33. R. Sethi and A. Roy-Chowdhury. Modeling and Recognition of Complex Multi-Person Interactions in Video. In *ACM Intl. Workshop on Multimodal pervasive video analysis*, 2010.
34. K. Shafique and M. Shah. A Non-Iterative Greedy Algorithm for Multi-Frame Point Correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, January 2005.
35. B. Song, T. Jeng, E. Staudt, and A. Roy-Chowdhury. A Stochastic Graph Evolution Framework for Robust Multi-Target Tracking. In *Euro. Conference on Computer Vision*, 2010.
36. B. Song, A. Kamal, C. Soto, C. Ding, J. Farrell, and A. Roy-Chowdhury. Tracking and Activity Recognition Through Consensus in Distributed Camera Networks. *IEEE Trans. on Image Processing*, October 2010.
37. B. Song and A. Roy-Chowdhury. Stochastic Adaptive Tracking in a Camera Network. In *IEEE Intl. Conf. on Computer Vision*, 2007.
38. B. Song and A. Roy-Chowdhury. Robust Tracking in A Camera Network: A Multi-Objective Optimization Framework. *IEEE Journal on Selected Topics in Signal Processing: Special Issue on Distributed Processing in Vision Networks*, 2008.
39. C. Soto, B. Song, and A. Roy-Chowdhury. Distributed Multi-Target Tracking In A Self-Configuring Camera Network. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

40. C. Stauffer and W.E.L. Grimson. Adaptive Background Mixture Models for Real-time Tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.
41. G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
42. J. Xiao, H. Cheng, H. S. Sawhney, and F. Han. Vehicle Detection and Tracking in Wide Field-of-View Aerial Video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.
43. J. Xing, H. Ai, and S. Lao. Multi-Object Tracking Through Occlusion by Local Tracklets Filtering and Golbal Tracklets Assiciation with Detection Responses. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
44. M. Yang, Y. Wu, and G. Hua. Context-Aware Visual Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, July 2009.
45. Q. Yu, G. Medioni, and I. Cohen. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
46. L. Zhang, Y. Li, and R. Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.