A Stochastic Graph Evolution Framework for Robust Multi-target Tracking*

Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury

Dept. of Electrical Engineering, University of California, Riverside, CA 92521, USA

Abstract. Maintaining the stability of tracks on multiple targets in video over extended time periods remains a challenging problem. A few methods which have recently shown encouraging results in this direction rely on learning context models or the availability of training data. However, this may not be feasible in many application scenarios. Moreover, tracking methods should be able to work across different scenarios (e.g. multiple resolutions of the video) making such context models hard to obtain. In this paper, we consider the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. We build our solution on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets). By analyzing the statistical properties of these tracklets, we develop associations between them so as to come up with longer tracks. This is achieved through a stochastic graph evolution step that considers the statistical properties of individual tracklets, as well as the statistics of the targets along each proposed long-term track. On multiple real-life video sequences spanning low and high resolution data, we show the ability to accurately track over extended time periods (results are shown on many minutes of continuous video).

1 Introduction

Multiple object tracking is the most fundamental task for higher level automated video content analysis. Although a large number of trackers exist, stable, long-term tracking is still a challenging problem. Common reasons which cause tracking failure are occlusion, illumination change, clutter and sensor noise. 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. Therefore, detection and correction of the errors in the tracks is the key to robust long term tracking.

Many state-of-the-art tracking algorithms focus on how to avoid losing track. They usually rely on training data or learning context models (e.g. some recent papers like [1,11,16]). In many situations, there may not be enough data for training or learning context models. For example, videos downloaded from

* 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).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 605–619, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Youtube are usually a few minutes in length and from a variety of contexts. Analysis of these videos requires tracking and there is no separate data available to learn models.

In this paper, we consider the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. *We are not proposing our method as an alternative to learning models, rather as an approach for applications where such data is not available.* Building on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets), we propose a stochastic graph evolution framework to understand the association between tracklets so as to come up with longer tracks by analyzing the statistical properties of individual tracklets, as well as the statistics of the targets along each proposed long-term track.

Our approach is original in the following ways.

– We come up with a measure of the accuracy of the tracking, so that we can determine when the tracking error is increasing and identify the tracklets.
– We propose a prediction-based affinity modeling approach by searching for optimal associations in the target feature space using a stochastic sampling method. We show that this provides higher accuracy as opposed to heuristically selecting a fixed affinity model. This process leads to a weighted graph with the tracklets as nodes and affinity scores as weights.
– We consider long-term interdependencies between the target tracklet features and use it to correct for wrong correspondences. This is achieved by evolving the graph weights through a stochastic sampling approach. The underlying hypothesis for this step is that along a correct track the variation of the target features will be lower than along a wrong track.

Through this process, we are able to get stable long-term tracks of multiple targets without the need for extra training data. Our method analyzes the video in a time-window (maximum duration of a few minutes) in a batch process; thus there is a delay in the analysis, which is often a non-issue in many applications.

1.1 Related Work

To track multiple objects, a lot of effort has been devoted to making data association based on the results of object detection. Multi-Hypothesis Tracking (MHT) [13] and Joint Probabilistic Data Association Filters (JPDAF)[2] are two representative methods. In order to overcome the large computational cost of MHT and JPDAF, various optimization algorithms such as Linear Programming [9], Quadratic Boolean Programming [10], and Hungarian algorithm [12] are used for data association. In [17], data association was achieved through a MCMC sampling based framework. These methods rely on the precision of object detection, which can not be guaranteed in complex scenarios. On the other hand, some statical tracking methods (e.g. Kalman filter and particle filter [8]) and kernel tracking algorithm (e.g. mean-shift tracker [3]) release the requirement for object detection in every frame, but they are not powerful for tracking

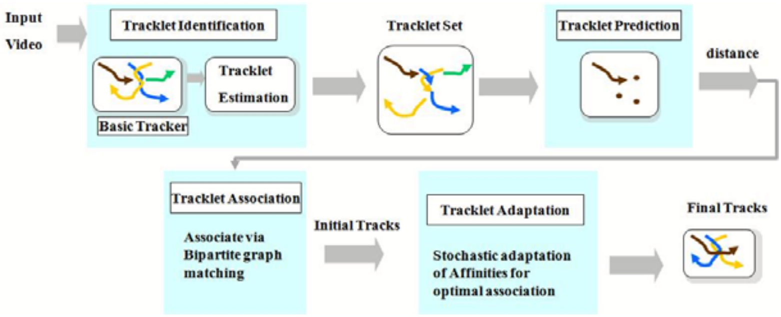


Fig. 1. Overview of proposed approach

multiple objects by themselves. In [7], particle filters were used to track multiple objects by incorporating probabilistic MHT for data association.

Many state-of-the-art tracking algorithms focus on how to avoid errors in tracking. In [18], the authors proposed a min-cost flow framework for global optimal data association. A tracklet association based tracking method was presented in [5], which fixed the affinity model heuristically and focused on searching for optimal associations. A HybridBoosted affinity model was learned in [11]. The method is built on the availability of training data under a similar environment, which may not be always feasible. The authors in [1] addressed the problem of learning an adaptive appearance model for object tracking. Context information was considered in [16] to help in tracking, by integrating a set of auxiliary objects which are learned online. Unfortunately, except for high resolution video, it is not easy to find these auxiliary objects.

We would like to clearly differentiate our approach with traditional Data Association Tracking (DAT) approaches which perform the tracking by detection instead of running a tracking algorithm. Unlike the DAT methods, our data association is done on the tracking results, not the detection result. Moreover, in most methods, there is very little attention paid on error recovery, i.e., if errors happen, how to detect and correct them. It is, however, at the heart of the proposed strategy.

2 Overview of Solution Strategy

Our system is initialized when new targets are detected. A basic tracker using particle filter is applied to generate the initial tracks. It can be replaced by any existing tracker, without affecting the other modules. However, errors cannot be avoided in the tracks generated by the basic trackers, especially in the presence of occlusions, disappearance of targets and close proximity of targets. In order to correct the errors, we propose a stochastic tracklet association and adaptation strategy.

Fig. 1 shows an overview of our long-term tracking system. We begin 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. Details on estimation of tracklets are provided in Section 3.

The tracklets are then associated based on their affinities. Although an optimal affinity model could be learned [11], it requires the availability of training data. Instead of using a heuristically selected fixed affinity model, we propose a prediction based affinity modeling approach by searching for optimal predictions in the feature space based on Markov chain Monte Carlo (MCMC) sampling methods as detailed in Section 4. The tracklets are first extended in space and time through new predicted positions generated using the Metropolis Hastings algorithm. The affinity between two tracklets is modeled by the distance (in a suitable feature space) of the predicted ending of one tracklet to the starting of another. Using the affinity model, we create a tracklet association graph (TAG) 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. This is explained in Section 4.1.

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. The prediction based affinity model may not be enough to capture the variation. This leads us to develop a graph evolution scheme as described in Section 5. 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. We design a loss function and an efficient optimization strategy for this process. The overall approach is able to track stably over minutes of video in challenging domains with no learning and context information.

3 Tracklet Identification

As mentioned earlier, we identify the tracklets from the initial tracks generated from the basic tracker. Then the problem of tracking over long-term video is equivalent to finding the best association between the tracklets. Note that although the particle filter based basic tracker is replaceable, it was chosen because the observation model is nonlinear and the posterior can temporarily become multimodal due to background clutter. We now describe our implementation of the basic tracker using a particle filter and the tracklet estimation scheme.

3.1 Particle Filter Based Basic Tracker

Initialization: We use motion detection to automatically detect moving objects. The background modeling algorithm in [15] is used for its adaptability to illumination change, and to learn the multimodal background through time. Using the learned background model, the moving objects can be detected. However, the background model may not be precise due to noise, which could produce false detections. By observing that most of our interested targets, like people and vehicles, are on ground plane, we estimate the rough ground plane area using the method proposed in [6]. Based on the ground plane information, false alarms can be removed significantly. We reiterate that this process is just one choice based on the current literature. It can be replaced and we do not assume that this step should work perfectly. In fact, the following stages are designed to correct for the errors here.

System model: The target regions are 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. We consider a linear dynamic model: $X_t = AX_{t-1} + n_t$,

where A defines the deterministic system model and n_t is zero mean white Gaussian noise ($n_t \sim \mathcal{N}(0, \Sigma_t)$).

Observation model: 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. Our observation models were generated by combining an appearance and a foreground response model, i.e.,

$$p(I_t|X_t) = p(I_t^a, I_t^f|X_t), \tag{1}$$

where I_t^a is the appearance information of I_t and I_t^f is the foreground response of I_t using the learned background model as described above. 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 (1) becomes $p(I_t|X_t) = p(I_t^a|X_t)p(I_t^f|X_t)$. The appearance observation likelihood is 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 is $p(I_t^f|X_t) \propto \exp\{-(1-\frac{\#F(X_t)}{\#X_t})^2\}$, 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.

3.2 Tracklet Estimation

Errors cannot be avoided in the tracks generated by any basic tracker. 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 leads us to the rules for tracklet estimation. We estimate when these errors happen and identify their spatio-temporal location, leading to the tracklets. An example is shown in Fig. 2.

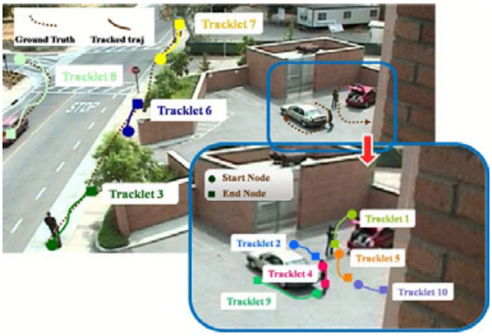


Fig. 2. An example of tracklet identification. The ground truth trajectories are represented by brown dotted lines. The estimated tracklets due to detection of a lost track (track of the person in lower left corner due to occlusion) and targets' close proximity (the persons moving around the cars) are clearly shown in different colors.

Detection of lost track: The tracking error (TE) [2] 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. In our observation model, TE of tracked target $\hat{X}_t$ is calculated by

$$TE(\hat{X}_t, I_t) = TE_a(\hat{X}_t, I_t) + TE_f(\hat{X}_t, I_t), \tag{2}$$

where $TE_a(\hat{X}_t, I_t) = B(ch(X_t), ch_0)^2$ 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 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 inspect the distances between targets, and break the tracks into tracklets at the points where targets are getting close, as shown in Fig. 2.

4 Prediction Based Tracklet Affinity Modeling

As mentioned in [11], in most previous work, simple affinity models are used by heuristically selecting parameters. The approach in [11] is able to automatically select among features and corresponding non-parametric models based on training data. However, without the availability of training data, searching in such an affinity function space is not trivial. Under this condition, rather than directly search in the affinity function space, we propose a prediction based affinity modeling approach by searching for optimal predictions in the feature space based on MCMC sampling methods and using the predicted features to come up with the affinity measurements. This provides more robustness compared to using a fixed affinity measure, as shown in Table 3 in Section 6.

4.1 Tracklet Prediction and Association

The tracklet occurring earlier in time is referred to as the base-tracklet, while a tracklet beginning after the base-tracklet ended is referred to as the target-tracklet. In order to measure tracklet affinity, the base-tracklet is extended in the image motion/appearance-space M steps, where M could represent the number of frames that separate the end of the base-tracklet from the beginning of the target-tracklet or a fixed number of pre-determined steps. In order to choose new points for the base-tracklet, a form of MCMC called the Metropolis Hastings Algorithm is used to generate chains of random samples.

MCMC is a versatile tool for generating random samples that can be used in determining statistical estimates. By using this sampling method, the algorithm is able to take advantage of the base object's motion and appearance information while also considering its relationship to the target-tracklet via the target distribution $p_{tl}(\mathbf{z})$. The target distribution relates points surrounding the starting point of the target-track to a probability measure. MCMC has the advantage of not requiring perfect knowledge of the target distribution $p_{tl}(\mathbf{z})$ – it is enough to be able to evaluate it a particular point, but not sample from it.

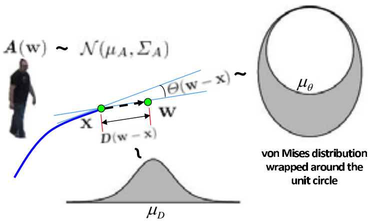

$A(\mathbf{w}) \sim \mathcal{N}(\mu_A, \Sigma_A)$
$\Theta(\mathbf{w}-\mathbf{x}) \sim$
$\mathbf{x}$
$\mathbf{w}$
$D(\mathbf{w}-\mathbf{x})$
$\sim$
μ_D
μ_θ
von Mises distribution wrapped around the unit circle

Fig. 3. An illustration of proposing a new point based on the proposal distribution

The Proposal Distribution: The proposal distribution $q_{tl}(\mathbf{y}|\mathbf{z})$ allows us to generate samples from a distribution that is easy to sample from. Our proposal distribution was based on a combination of motion and appearance of each target. The direction of motion of each target is modeled using the von Mises distribution. The von Mises distribution has close ties to the normal distribution, however it is limited to angles about the unit circle as shown in Fig. 3. The pdf for the von Mises distribution takes the following form:

$$v(\theta|\mu_\theta, \kappa) = \frac{e^{\kappa \cos(\theta - \mu_\theta)}}{2\pi I_0(\kappa)}. \tag{3}$$

Here, $I_0(.)$ is the modified Bessel function of order zero. The parameters μ_θ and κ correspond to mean and variance in a normal distribution, which are learned within each base tracklet.

The speed of each target is modeled with a Normal distribution $\mathcal{N}(\mu_D, \sigma_D)$, where the mean μ_D and variance σ_D are learned within each base tracklet. The appearance model is described using a normal distribution on the color histogram of each target as $\mathcal{N}(\mu_A, \Sigma_A)$, where the parameters are also learned within each base tracklet.

So our proposal distribution is

$$q_{tl}(\mathbf{w}|\mathbf{x}) \propto v(\Theta(\mathbf{w}-\mathbf{x})|\mu_\theta, \kappa)\mathcal{N}(D(\mathbf{w}-\mathbf{x})|\mu_\mathcal{D}, \sigma_\mathcal{D})\mathcal{N}(A(\mathbf{w})|\mu_A, \Sigma_A), \qquad (4)$$

where $\Theta(\mathbf{w}-\mathbf{x})$ and $D(\mathbf{w}-\mathbf{x})$ represent the angle and distance between the proposed point $\mathbf{w}$ and the end point of tracklet $\mathbf{x}$ respectively, and $A(\mathbf{w})$ represents the color histogram of proposed point $\mathbf{w}$. A new point is proposed by randomly producing motion direction, speed and appearance vector as shown in Fig. 3.

The Target Distribution: Proposed points from the base-tracklet were related to the starting point of the target-tracklet through the target distribution. The target distribution, $p_{tl}(\mathbf{z})$, was chosen as

$$p_{tl}(\mathbf{z}) \propto e^{-d_\mathbf{z}}, \qquad (5)$$

where $d_\mathbf{z} = \sqrt{d_a^2 + d_m^2}$ is a Euclidean combination of the normalized distance in the motion-space, d_m, and the Bhattacharyya distance, d_a, between the image histograms of the average base and target appearances.

M-H Algorithm: Given the proposal distribution, $q_{tl}(\mathbf{w}|\mathbf{x})$, where $\mathbf{w}$ was the proposed point and $\mathbf{x}$ was the last point in the tracklet and the target distribution $p_{tl}(\mathbf{w})$, the probability that a point was accepted was given as,

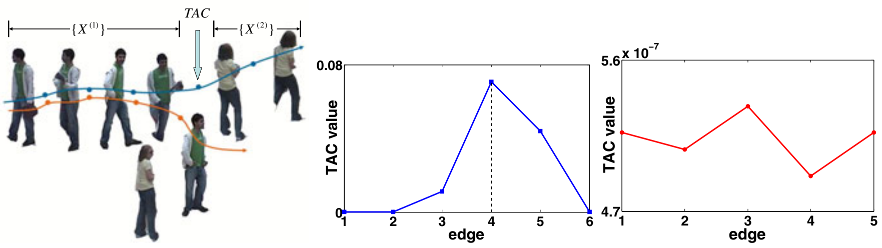
$$\rho_{tl}(\mathbf{x}, \mathbf{w}) = \min\left\{\frac{p_{tl}(\mathbf{w})q_{tl}(\mathbf{x}|\mathbf{w})}{p_{tl}(\mathbf{x})q_{tl}(\mathbf{w}|\mathbf{x})}, 1\right\}. \qquad (6)$$

This process results in a sequence of accepted points for M time steps. The affinity between a base and target tracklet is computed as the distance $d_\mathbf{z}$ in (5) between the end of the predicted extension of the base tracklet and the beginning of the target tracklet.

Tracklet Association We can now define a Tracklet Association Graph where the nodes are the identified tracklets and the weights on the edges are the affinity scores. By splitting the beginning and end of each tracklet into two subsets, the problem of the tracklet association can be formulated as a maximum matching problem in a weighted bipartite graph. In this paper, we use the Hungarian algorithm [12] to find the maximum matching.

5 Tracklet Adaptation

If the affinity scores (edge weights) of the bipartite graph were known exactly and assumed to be independent, the tracking problem could be solved optimally by the tracklet association method described above. However, it is not uncommon for some of the similarities to be estimated wrongly since they depend on detected features which is not a perfect process. As we show in Fig. 4, if the similarity



(a) (b) (c)

Fig. 4. (a) Tracklets of two targets obtained from Videoweb courtyard dataset of Section 6: 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.

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.

We address this issue by constructing a graph evolution strategy, in which the weights (i.e., affinity scores) on the edges of the tracklet association graph are adapted by measuring the similarity of observed features along a path that is generated after tracklet association. We adopt the affinity adaptation method proposed in [14], but instead of adapting deterministically which may be stuck at a local optimum, we propose a Metropolis-Hastings based adaptation scheme with the potential to reach the global optimal.

5.1 Tracklet Association Cost Function

To model the spatio-temporal variation of the observed features along a path, a Tracklet Association Cost (TAC) is defined motivated by [14]. Given an estimated track for the q^{th} target, λ_q, TAC is defined on each edge $e_{ij} \in \lambda_q$. The feature vector of the tracklets before (in time) e_{ij} on λ_q and those after e_{ij} are treated as two clusters. An illustration of TAC calculation is shown in Fig. 4 (a).

Let $\{X\}$ be the set of feature (e.g., appearance) of all N tracklets along the path and let them be clustered into $\{X^{(1)}\}$ and $\{X^{(2)}\}$ with respect to each edge $e_{ij} \in \lambda_q$. Let the mean m of the features in $\{X\}$ be $m = \frac{1}{N}\sum_{x\in\{X\}} x$. Let m_i be the mean of N_i data points of class $\{X^{(i)}\}$, $i = 1, 2$, such that $m_i = \frac{1}{N_i}\sum_{x\in\{X^{(i)}\}} x$. Let S_T be the variance of the all observed feature x along the path, i.e., $S_T = \sum_{x\in\{X\}} |x-m|^2$ and S_W be the sum of the variances along each sub-path, $\{X^{(1)}\}$ and $\{X^{(2)}\}$, i.e., $S_W = \sum_{i=1}^{2} S_i = \sum_{i=1}^{2}\sum_{x\in\{X^{(i)}\}} |x - m_i|^2$.

The TAC for e_{ij} is defined as

$$TAC(e_{ij}) = \frac{|S_T - S_W|}{|S_W|} \triangleq \frac{|S_B|}{|S_W|}. \qquad (7)$$

Thus the TAC is defined from Fisher's linear discriminant function [4] and measures the ratio of the distance between different clusters, S_B, over the distances between the members within each cluster S_W. If all the feature nodes along a path belong to the same target, the value of TAC at each edge $e_{ij} \in \lambda_q$ should be low, and thus the variance of TAC over all the edges along the path should also be low. If the feature nodes belonging to different people are connected wrongly, we will get a higher value of TAC at the wrong link, and the variance of TAC along the path will be higher. Thus, the distribution of TAC along a path can be used to detect if there is a wrong connection along that path.

We can now design a loss function for determining the final tracks by analyzing features along a path. We specify the function in terms of the Tracklet Association Cost (TAC) function. Thus, we adapt the affinity scores to minimize

$$L(\lambda_q) = \sum_{\lambda_q} Var(TAC(e_{ij} \in \lambda_q^{(n)})). \tag{8}$$

5.2 Metropolis-Hastings Based Adaptation of Tracklet Association

Whenever there is a peak[1] in the TAC function for some edge along a path, the validity of the connections between the features along that path is under doubt. As per the Metropolis-Hastings method, we will propose a new candidate affinity score s'_{ij} on this edge where the peak occurs using a proposal distribution $q_{af}(s'_{ij}|s_{ij})$, where s_{ij} is the affinity score on edge e_{ij}. The proposal distribution $q_{af}(s'_{ij}|s_{ij})$ is chosen to be an uniform distribution of width 2δ, i.e., $U(s_{ij} - \delta, s_{ij} + \delta)$, since without additional information, uniform distribution can be a reasonable guess of the new weights. Any other distribution can be chosen based on the application.

We then recalculate the maximum matching paths, λ'_q, of the new feature graph. The target probability $p_{af}(.)$ is defined as $p_{af}(s_{ij}) \propto \exp(-L(\lambda_q))$, and $p_{af}(s'_{ij}) \propto \exp(-L(\lambda'_q))$. The candidate weight s'_{ij} is accepted with probability $\rho_{af}(s_{ij}, s'_{ij})$ as

$$\rho_{af}(s_{ij}, s'_{ij}) = \min\left\{ \frac{p_{af}(s'_{ij}) q_{af}(s_{ij}|s'_{ij})}{p_{af}(s_{ij}) q_{af}(s'_{ij}|s_{ij})}, 1 \right\}. \tag{9}$$

Our adaptation scheme is summarized below.

1. Construct a weighted graph $G = (V, E, S)$, where the vertices are the tracklets and edge weights are set as described in Section 4.
2. Estimate the optimal paths, $\tilde{\lambda}_q$ based on bipartite graph matching.
3. Compute the TAC for each $e_{ij} \in \tilde{\lambda}_q$.
4. Propose a weight s'_{ij} on the link where the TAC peak occurs based on a proposal distribution.

[1] The peak is detected if it is above a threshold, which is defined as $E\{TAC(e_{ij} \in \lambda_q)\} + 2\sqrt{Var(TAC(e_{ij} \in \lambda_q))}$.

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

5. Recalculate the maximum matching paths, λ'_q, of the new feature graph. We accept the new graph with probability $\rho_{af}(s_{ij}, s'_{ij})$ in (9).
6. Repeat Steps 4 and 5 until either a predefined iteration number is reached or the system reaches some predefined stopping criterion.

6 Experimental Results

To evaluate the performance of our system, we show results on two different data sets. The CAVIAR (http:// homepages.inf.ed.ac.uk/rbf/CAVIARDATA1) is captured in a shopping mall corridor with heavy inter-object occlusion. The Videoweb dataset (http://vwdata.ee.ucr.edu) is a wide area multi-camera dataset consisting of low and high resolution videos. We consider two subsets of videos. The first is a outdoor low resolution parking lot scene, and the second is a relatively high resolution courtyard scene with intensive occlusion and clutter.

To evaluate the performance of our system quantitatively, we adopt the evaluation metrics for tracking defined in [11] and [18]. In addition, we define RS and RL to evaluate the ability of recovering from occlusion (see Table 1). Although we show results on datasets that others have worked with, it should be noted that we are not proposing our method as an alternative to those that use/learn context models, rather as an approach to be used when such models are not available. Therefore, our results should be analyzed with the ground truth, rather than against those that rely on such knowledge.

Results on CAVIAR dataset: 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. We show our results on the relatively more challenging part of the dataset which contains 7 videos (TwoEnterShop3, TwoEnterShop2, ThreePastShop2, ThreePastShop1, TwoEnterShop1, OneShopOneWait1, OneStopMoveEnter1)[2]. Table 2 shows the comparison among the proposed method, the min-cost flow approach in [18], HybridBoosted affinity

[2] Compared with other sequences in CAVIAR (e.g. TwoLeaveShop2, OneStopNoEnter1 and OneStopMoveNoEnter1), the challenge of the set we test on is obvious.

Table 2. Tracking Results on CAVIAR data set. Results of [11] and [18] are reported on 20 sequences; basic particle filter and proposed method are reported on 7 most challenging sequences of the dataset. Our test data has totally 12308 frames for about 500 sec.

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

Table 3. Tracking Results on one sequence of CAVIAR dataset. Proposed approach is a combination of basic particle filter, prediction based affinity model and track adaptation.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle fitler	18	44.4%	22.2%	7	6	4/14	0/5
Simple Affinity model	18	66.6%	5.6%	2	4	12/14	2/5
Prediction-Based Affinity model	18	72.2%	0.0%	2	3	13/14	3/5
Proposed method	18	83.3%	0.0%	2	1	13/14	4/5

modeling approach in [11] and a basic particle filter. The results in [11,18] are reported on 20 sequences in CAVIAR. It can be seen that our method achieves similar performance as in [11,18]. It should also be noted that [11,18] are built on the availability of training data under similar environment (e.g. other 6 sequences in CAVIAR are used for training in [18]), while our method does not rely on any training; also our results are for the most challenging sub-part of the dataset. Some sample frames with results are shown in Fig. 5 (a). In the supplementary material, we show results on continuously tracking this data.

In order to show the achievement of each step (i.e., the prediction based affinity modeling and tracklet adaptation) of our proposed method, we compare the performances of the basic particle filter, a simple affinity model followed by bipartite graph match, prediction based affinity model without tacklet adaptation step, and the complete proposed approach on one of the sequences (the one shown in the supplementary material). The simple affinity model is constructed by directly using the average angle and speed of motion and average color histogram similar to [18]. It is clearly shown in Table 3 that our method has much less Fragments (FG) and ID Switches (IDS) and the adaptation part can further correct the wrong connections.

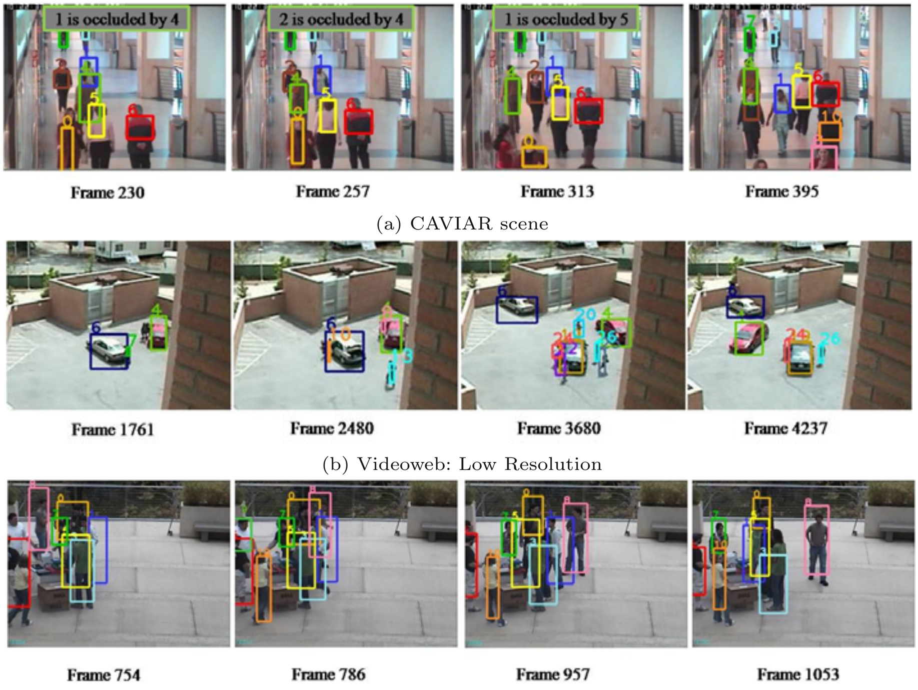
Results on Videoweb dataset – Low-resolution Example: The first part of Videoweb dataset we use is a low resolution parking lot scene. The target categories include people, cars and motorcycle (any object which is below 15 pixels in width is not taken into account). The low resolution makes tracking more challenging, especially in outdoor scenes since the illumination is always unstable and the appearance is hard to extract. The results of our methods are shown in Table 4. Some sample frames and tracking results are shown in 5 (b).

Table 4. Tracking Results on parking lot scene of Videoweb dataset. 4 sequences of totally 14673 frames (980 sec.) were used.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle filter	90	80%	10.0%	20	6	5/19	1/8
Proposed method	90	90%	4.4%	8	3	15/19	5/8

Table 5. Tracking Results on courtyard scene of Videoweb dataset, 4 sequences of totally 8254 frames (550 sec.) were used.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle fitler	48	41.7%	14.6%	9	17	10/35	2/15
Proposed method	48	66.7%	6.25%	5	8	29/35	12/15



(a) CAVIAR scene

(b) Videoweb: Low Resolution

(c) Videoweb: High Occlusion and Clutter

Fig. 5. (a): Tracking results on CAVIAR dataset. (b): Tracking results on Videoweb dataset - low resolution parking lot scene. (c): Tracking results on Videoweb dataset - high clutter and occlusion courtyard scene.

Results on Videoweb dataset – High Occlusion and Clutter Example: The second part of Videoweb dataset consists of multiple people interacting in a courtyard. It is almost impossible to track with a basic tracker because of very high occlusion. Also, an adaptive background model is hard to build for this level of occlusion. The tracking result shows our method using the proposed strategy can get reasonable results even at this level of occlusion. The performance on this dataset is shown in Table 5. Some sample frames with tracking results are shown in 5 (c). Results on tracking about 45 seconds of this scene are shown in the supplementary material.[3]

7 Conclusions

In this paper, we considered the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. We built our solution on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets). We then developed associations between them so as to come up with longer tracks. Finally, we proposed a graph evolution method to search for optimal association, then providing robustness to inaccuracies in feature similarity estimation. Promising results are shown on challenging data sets.

References

1. Babenko, B., Yang, M., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: IEEE CVPR (2009)
2. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
3. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence (May 2003)
4. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley Interscience, Hoboken (2001)
5. Ge, W., Collins, R.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In: British Machine Vision Conference (2008)
6. Hoiem, D., Efros, A., Hebert, M.: Geometric Context from a Single Image. In: IEEE ICCV (2005)
7. Hue, C., Cadre, J.L., Prez, P.: Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion. IEEE Trans. on Signal Processing (2002)
8. Isard, M., Blake, A.: Condensation - Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision (1998)
9. Jiang, H., Fels, S., Little, J.: A Linear Programming Approach for Multiple Object Tracking. In: IEEE CVPR (2007)
10. Leibe, B., Schindler, K., Gool, L.V.: Coupled Detection and Trajectory Estimation for Multi-Object Tracking. In: IEEE ICCV (2007)
11. Li, Y., Huang, C., Nevatia, R.: Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. In: IEEE CVPR (2009)
12. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. In: IEEE CVPR (2006)

[3] More extensive tracking results are available on the author's webpage.

13. Reid, D.: An Algorithm for Tracking Multiple Targets. IEEE Trans. Automatic Control 24(6), 843–854 (1979)
14. Song, B., Roy-Chowdhury, A.: Stochastic Adaptive Tracking in a Camera Network. In: IEEE ICCV (2007)
15. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Real-time Tracking. In: IEEE CVPR (1998)
16. Yang, M., Wu, Y., Hua, G.: Context-Aware Visual Tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence (July 2009)
17. Yu, Q., Medioni, G., Cohen, I.: Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In: IEEE CVPR (2007)
18. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: IEEE CVPR (2008)