Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association

Qian Yu Gérard Medioni
Institute of Robotics and Intelligent Systems
University of Southern, Los Angeles, CA 90089
{qianyu,medioni}@usc.edu

Isaac Cohen
Automation & Control Solutions
Honeywell, Minneapolis, MN 55418
isaac.cohen@honeywell.com

Abstract

We propose a framework for general multiple target tracking, where the input is a set of candidate regions in each frame, as obtained from a state of the art background learning, and the goal is to recover trajectories of targets over time from noisy observations. Due to occlusions by targets and static objects, noisy segmentation and false alarms, one foreground region may not correspond to one target faithfully. Therefore the one-to-one assumption used in most data association algorithm is not always satisfied. Our method overcomes the one-to-one assumption by formulating the visual tracking problem in terms of finding the best spatial and temporal association of observations, which maximizes the consistency of both motion and appearance of trajectories. To avoid enumerating all possible solutions, we take a Data Driven Markov Chain Monte Carlo (DD-MCMC) approach to sample the solution space efficiently. The sampling is driven by an informed proposal scheme controlled by a joint probability model combining motion and appearance. To make sure the Markov chain to converge to a desired distribution, we propose an automatic approach to determine the parameters in the target distribution. Comparative experiments with quantitative evaluations are provided.

1. Introduction

Multiple targets tracking is a critical component of video surveillance systems, as it provides the description of spatio-temporal relationships among moving objects in the scene required by activity recognition modules. Under a general tracking setup, environments of interest usually contain an unknown number of targets and multiple observations of targets are reported. The purpose of data association in multiple target tracking problem is to recover the correct correspondence between observations and targets. Once data association is established, filtering techniques are applied to estimate the state of targets.

Most existing data association algorithms consider a one-to-one mapping between targets and observations, which assumes that at a given time instant one observation can be associated with at most one target and vice versa: one target corresponds to at most one observation. This assumption is reasonable when the considered observations are punctual. However, in the visual tracking problem, the observations correspond to blobs or meaningful regions which cannot be faithfully modeled by a single point. Moreover, erroneous detections due to occlusion and spurious motion segmentation provide a set of observations where a single moving object is often detected as multiple moving regions, or multiple moving regions are merged into a single blob. Therefore, the one-to-one association is usually violated in real environments. In this paper, we propose a general framework which makes use of spatio-temporal consistency in both motion and appearance and does not require the one-to-one mapping between observations and targets. Although our framework can accommodate additional information, such as generic model information (which is discussed in future work), in this paper, we use consistency of motion and appearance as the only constraint.

Instead of inferring the association and targets' states according to current observations, our method uses a batch of observations. A track is regarded as a path, in space-time, traveled by a target. We aim to recover the tracks of an unknown number of targets using the consistency in motion and appearance of tracks. Due to the high computational complexity of such an association scheme, a Data-Driven Markov Chain Monte Carlo (DD-MCMC) [15] method is proposed to sample the solution space. Both spatial and temporal association samples are incorporated into the Markov chains' transitions. The key contribution of the paper is the explicit use of spatio-temporal smoothness in motion and appearance to overcome the one-to-one assumption used in most of data association algorithms by using a spatio-temporal MCMC.

The paper is organized as follows. The related work is

1-4244-1180-7/07/$25.00 ©2007 IEEE

reviewed in Section 2. We formulate the multiple target tracking problem and present our spatio-temporal MCMC data association algorithm in Section 3 and 4 respectively. We discuss how to determine the parameters used in our probabilistic model by Linear Programming and provides comparative results on both simulation and real data sets in Section 5, followed by conclusions and discussion in Section 6.

2. Related Work

In the past decades, multiple target tracking has been a very active field. Among the large body of work, the multiple hypothesis tracker (MHT) [13] and joint probabilistic data association filter (JPDAF) [2], are the most widely used. MHT is a statistical framework to evaluate the likelihood of each hypothesis, which represents a set of assignments of observations and targets. To find the best hypothesis over time, in practice k-best hypotheses are maintained at each time, which can be solved in polynomial time [5]. The essential difference between JPDAF and MHT is that instead of finding the best hypothesis, JPDAF computes expectation of the state of targets over all hypotheses (joint association events). Any practical implementation of these algorithms requires pruning all hypotheses to a smaller hypotheses set. Both these data association methods assume the one-to-one mapping between observations and targets.

According to the depth of observations, the existing methods can be categorized into either sequential inference or deferred logical inference. Sequential methods make inference according to current observations. In [8], a MCMC-based particle filter simulates the distribution of the association probability with a fixed number of targets, which allows multiple temporal associations between observations and targets. In [7, 14], sequential tracking methods use pairwise Markov random field (MRF) based prior to model the interaction between targets at one time instant. In [16], Markov network is used to model the interaction between multiple targets at each time and a mean field Monte Carlo algorithm is applied to approximately estimate the posterior density of each target. In [17], multiple people are detected and tracked in crowded scene using MCMC based method to estimate the state and the number of targets sequentially. In [10] a multi-view approach uses particle filter based method to segment and track people against clutter. Many sequential methods employ model information to identify a specific type of target against background, such as [16, 17, 8, 7].

Due to the ambiguity existing at any one time instant, deferred logical inference makes a decision according to a sequence of observations. By extending a hypothesis from an assignment set between observations and targets at one time to a set of disjoint tracks, both MHT and JPDAF have a deferred logical version. However, the solution space of hypotheses grows exponentially in term of the depth of observations. In [11], the authors adopted a deferred logic framework to organize the one-to-one temporal association between targets and punctual observations over long sequences. A dynamic programming method is applied in [3] to optimize trajectories with a sliding window.

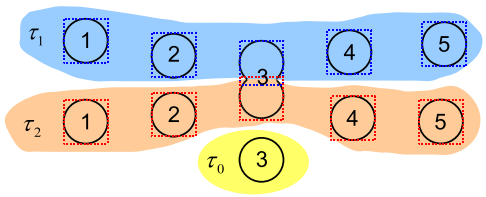


Figure 1. One possible cover of the observations, which includes two tracks (τ_1, τ_2) and one false alarm. The circles represent the foreground regions. The number in circles denotes the frame number of foreground regions. The dashed rectangles represent the covering rectangles of foreground regions

3. General Multiple Target Tracking

3.1. A Bayesian Formulation

Let $T \in \mathbb{N}^+$ denote the duration of tracking and assume that there are K unknown moving targets, within the time interval $[1, T]$. Our observations come from foreground regions. Let y_t denote the set of foreground regions at time t, Y is the set of all available foreground regions within $[1, T]$. Here, we define the tracking problem as given the observation Y, inferring an unknown number of K tracks in Eq. 1, where τ_0 is the set of false alarms, τ_k is the k^{th} track.

$$\omega = \{\tau_0, \tau_1, \cdots, \tau_K\} \tag{1}$$

Each τ_k in ω is defined as a sequence of shapes. For simplicity, we use rectangles to represent the shapes covering foreground regions. Therefore, tracking can be regarded as the problem to find a cover which maximizes spatio-temporal consistency of targets. In the case of a single target with perfect foreground segmentation, the set of MBRs (*Minimum Bounding Rectangles*) of each foreground region at different time forms the best cover of the target. However, when inter-occlusion between multiple targets and noisy foreground segmentation exist, it is not trivial to find the optimal cover. Figure 1 shows one possible cover of targets, which includes two tracks (τ_1, τ_2) and one false alarm.

In our framework, the tracking problem is formulated as maximizing a posterior (MAP) of a cover of foreground regions, given the set of observations Y:

$$\omega^* = \arg\max(p(\omega|Y)) \tag{2}$$

By introducing the concept of cover, we overcome the one-to-one assumption at each time instant, one foreground region can be covered by more than one target and one target can cover more than one foreground region as well.

In a Bayesian framework, the cover ω is denoted by a set of hidden variables. In the following sections, we discuss the prior and likelihood model used in our method.

$$\omega \sim p(\omega|Y) \propto p(Y|\omega)p(\omega) \tag{3}$$

3.2. Prior Probability

To find a cover with reasonable properties, we first define a prior model which considers the following criteria: we prefer long tracks with few false alarms. In addition, one track should have little overlap with other tracks. Accordingly, we adopt the prior probability of a cover ω as the product of the following terms.

$$p(\omega) = p(L)p(F)p(O) \tag{4}$$

1. Length of each track. We adopt an exponential model $p(L)$ of the length of each track. Let $|\tau_k|$ denote the length, i.e. the number of elements in τ_k.

$$p(L) = \prod_{k=1}^{K} \frac{1}{z_0} \exp(\lambda_0 |\tau_k|) \tag{5}$$

2. False alarms. Let F denote the size of τ_0. We adopt an exponential model $p(F)$ to penalize the number of false alarms.

$$p(F) = \frac{1}{z_1} \exp(-\lambda_1 F) \tag{6}$$

3. Spatial overlap between different tracks. We adopt an exponential model in Eq.7 to penalize overlap between different tracks, where $\Gamma(t)$ denotes the average overlap ratio of different tracks at time t.

$$\begin{aligned} p(O) &= \prod_{t=0}^{T} \tfrac{1}{z_3} \exp(-\lambda_2 \Gamma(t)) \\ \Gamma(t) &= \frac{\sum_{\tau_i(t)\cap\tau_j(t)\neq\emptyset} |\frac{\tau_i(t)\cap\tau_j(t)}{\tau_i(t)\cup\tau_j(t)}|}{|\tau_i(t)\cap\tau_j(t)\neq\emptyset|} \end{aligned} \tag{7}$$

3.3. Joint Likelihood $p(Y|\omega)$

Within a small time span, the appearance of foreground regions covered by each track τ_k is supposed to be coherent, and the motion of such a rectangle sequence should be smooth. Hence, we consider a probabilistic framework for incorporating two parts of likelihoods: motion likelihood L_M, appearance likelihood L_A. We represent the elements (rectangles) in track k as $(\tau_k(t_1), \tau_k(t_2), \ldots, \tau_k(t_{|\tau_k|}))$, where $t_i \in [1, T]$, and $(t_{i+1} - t_i) \geq 1$, since missing detection may happen. Given one cover, the motion and appearance likelihood of a target is assumed to be independent of other targets. The joint likelihood of a cover can be factorized in Eq.8.

$$\begin{aligned} p(Y|\omega) &= \prod_{k=1}^{K} L(\tau_K) = \prod_{k=1}^{K} \prod_{i=1}^{|\tau_k|-1} L\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) \\ &= \prod_{k=1}^{K} \prod_{i=1}^{|\tau_k|-1} L_M\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) L_A\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) \end{aligned} \tag{8}$$

3.3.1 Motion Likelihood

For each target, we consider a linear kinematic model:

$$\begin{aligned} \mathbf{x}_{t+1}^k &= A\mathbf{x}_t^k + \mathbf{w} \\ \mathbf{y}_t^k &= H\mathbf{x}_t^k + \mathbf{v} \end{aligned} \tag{9}$$

where $\mathbf{x}_t^k$ is the kinematic state vector, which includes the position (u, v), size $(w, h)$and the first order derivatives $(\dot{u}, \dot{v}, \dot{w}, \dot{h})$ in 2D image coordinates. Measurement $\mathbf{y}_t^k$ in Eq.9 corresponds to the position and size of $\tau_k(t)$ in 2D image coordinates . $\mathbf{w} \sim \mathcal{N}(0, Q)$, $\mathbf{v} \sim \mathcal{N}(0, R)$ are Gaussian process noise and observation noise. The motion likelihood for each track can be represented as follows.

$$\begin{aligned} &L_M\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) = p(\tau_k(t_{i+1})|\hat{\tau}_k(t_i)) \\ &= 1/\left((2\pi)^2 \det(P_i)^{1/2}\right) \exp(-\tfrac{1}{2}\mathbf{e}_i^T P_i^{-1}\mathbf{e}_i) \end{aligned} \tag{10}$$

where $\mathbf{e}_i = \tau_k(t_{i+1}) - \bar{\tau}_k(t_{i+1})$ and $P_i = H^T \bar{P}_{i+1} H + R$. Let $\bar{\tau}_k(t_i)$ and $\hat{\tau}_k(t_i)$ denote the prior and posterior estimates of τ_k at time t_i. $\bar{P}_i$ is the prior estimate of state covariance at time t_i. The details of updating the prior and posterior estimates in Kalman filters can be found in [9]. Note that, if missing detection occurs in τ_k at time t, or say there is no observation at time t for track k, the prior estimate is assigned to the posterior estimate.

3.3.2 Appearance Likelihood

In order to model the appearance of each detected region, we adopt the non-parametric histogram-based descriptor [6] to represent the appearance of image blobs. The appearance likelihood is defined as follows.

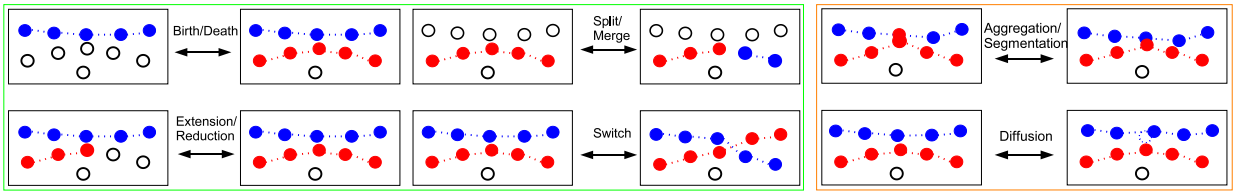
$$L_A\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) = (1/z_3) \exp\left(-\lambda_3 D(\tau_k(t_i), \tau_k(t_{i+1}))\right) \tag{11}$$

where $D(\cdot)$ represents the the symmetric Kullback- Leibler Distance (KL) between the histograms-based descriptor of foreground covered by $\tau_k(t_i)$ and $\tau_k(t_{i+1})$.

With some manipulations, we combine the prior $p(\omega)$ in Eq. 4 and the likelihood $p(\omega|Y)$ in Eq. 8 to have the whole posterior represented in Eq. 12.

$$\begin{aligned} &p(\omega|Y) \propto \exp\{-C_0 S_{len} - C_1 K - C_2 F - C_3 S_{olp} \\ &-C_4 S_{app} - S_{mot}\} \\ &S_{len} = -\left(\sum_{k=1}^{K} |\tau_k|\right), S_{olp} = \left(\sum_{T=1}^{T} \Gamma(t)\right) \\ &S_{app} = \sum_{k=1}^{K} \sum_{i=1}^{L_k - 1} D(\tau_k(t_i), \tau_k(t_{i+1})) \\ &S_{mot} = \sum_{k=1}^{K} \sum_{i=1}^{L_k} \left(\log\left(\det(P_i)^{1/2}\right) + \tfrac{1}{2}\mathbf{e}_i^T P_i^{-1}\mathbf{e}_i\right) \end{aligned} \tag{12}$$

where $C_0, \cdots, C_4$ are positive real constants. Eq. 12 reveals that the MAP estimation is equivalent to finding the minimum of an energy function. The tradeoff between prior and posterior will lead to a MAP solution. In the experiment's section, we discuss how to determine the parameters in such a probabilistic model by Linear Programming.



(a) Temporal Moves (b) Spatial Moves

Figure 2. Illustration of temporal and spatial moves. White circles represent false alarms.

4. Spatio-temporal MCMC Data Association

Searching in such a solution space for Eq.12 is not trivial. We propose to use a data-driven MCMC to estimate the best spatio-temporal cover of foreground regions. To ensure that detailed balance is satisfied, the Markov chain is designed to be ergodic and aperiodic . It is also important to design samplers that converge quickly. In our proposal distribution, the sampler contains two types of moves: temporal and spatial moves, shown in Figure 2. Temporal moves correspond to changing the labels of rectangles at different time instants, while spatial moves change covering rectangles at one time instant.

The overview of our MCMC data association algorithm is shown in Algorithm 1. The input to the algorithm is the set of original foreground Y, initial cover ω_0 and the total number of samples n_{mc}.

Each move is sampled according to its own prior probability. Since the temporal information is also applied in the spatial moves, we first take $\epsilon * n_{mc}$ ($\epsilon = 0.15$) temporal moves and then both types of moves are non-discriminatorily considered. Note that, instead of keeping all samples, we only keep the cover with the maximum posterior since we don't need the whole distribution but the MAP estimate. For the same reason, there is no burn-in procedure. The rectangles in the initial cover ω_0 are directly obtained from MBRs of foreground regions. Given the stationary distribution $\pi(\omega) = p(\omega|Y)$, the acceptance ratio $A(\omega, \omega')$ in Algorithm.1 is defined as follows.

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega|\omega')}{\pi(\omega)q(\omega'|\omega)}\right) \tag{13}$$

In such a Markov chain transition, the computation for each MCMC move is actually low, since we only need to compute the ratio $\pi(\omega')/\pi(\omega)$. To make the sampling more efficient, we define the neighborhood in spatio-temporal space. Two covering rectangles are regarded as neighbors if their temporal distance and spatial distance is smaller than a threshold. The neighborhood actually forms a graph, where a covering rectangle corresponds to a node. In the rest of the paper, we use "node" and "covering rectangle" interchangeably. The KD-tree structure is used to implement fast access to neighbors. A neighbor with a smaller (larger) frame number is called a parent (child) node. The neighborhood makes the algorithm more manageable since candidates are considered only within the neighborhood system. Figure 3 illustrates the neighborhood and $L\left(y|\tau_k(t_i)\right)$ represents the joint motion and appearance likelihood of assigning an observation y (*i.e.* one foreground region) to a track τ_k after t_i. In subsequent sections, we show how to devise the Markov chain's transition by considering specific choices for the proposal distribution $q(\omega'|\omega)$.

Algorithm 1 Spatio-temporal MCMC Data Association

Input: Y, n_{mc}, $\omega^* = \omega_0$ **Output**: ω^*
for $n = 1$ to n_{mc} **do**
  if $n < \epsilon * n_{mc}$ **then**
    Sample one temporal move.
  else
    Sample one move from all candidate moves.
  Propose ω' according to $q(\omega'|\omega)$
  Sample U from Unif$[0, 1]$
  if $U < A(\omega, \omega')$ **then** $\omega_n = \omega'$,
  else $\omega_n = \omega$
  if $p(\omega_n|Y) > p(\omega^*|Y)$ **then**
    $\omega^* = \omega_n$
end for

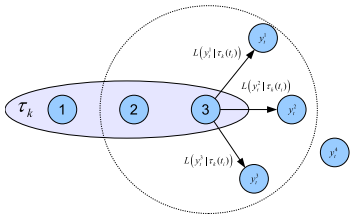


Figure 3. Neighborhood and association likelihood

4.1. Bi-directional sampling

Within the time span $[1, T]$, the "future" and "past" information is symmetric: *e.g*, we can extend a track in both the positive time direction and the opposite direction. Thus, we draw samples *uniformly at random* (u.a.r) in both temporal directions: looking forward and backward. This bidirectional sampling has more flexibility and reduces the total number of samples. In the following section, we only de-

scribe sampling in the positive time direction and the sampling in the other direction proceeds in a symmetric way.

4.2. Temporal moves

Forward Extension: First we u.a.r select a track to extend its length. We then select one false alarm node from the set of child nodes of the track's end node according to the joint likelihood $ex_k(y)$. We keep on extending the track according to a probability $\gamma \in [0, 1]$. Hence, the extension proposal distribution can be represented as follows.

$$q_{extension}(\cdot) = (1/K)p_T \prod_n \gamma ex_k(y)$$
$$ex_k(y) = \frac{(-\log L(y|\tau_k(t_i))+1)^{-1}}{\sum_{y \in child(\tau_k(t_i))} (-\log L(y|\tau_k(t_i))+1)^{-1}}$$

where n is the number of actual extensions. The prior of the extension move is p_T and all temporal moves have the same prior. **Forward Reduction:** We u.a.r select a track k to reduce its length. We then select a break point from $(|\tau_k| - 1)$ links according to the probability $br_k(i)$. The nodes in the track which are after the break point are moved to τ_0.

$$q_{reduction}(\cdot) = (1/K)p_T br_k(i)$$
$$br_k(i) = \frac{-\log L\big(\tau_k(t_{i+1})|\tau_k(t_i)\big)}{\sum_{j=1}^{|\tau_k - 1|} -\log L\big(\tau_k(t_{i+1})|\tau_k(t_i)\big)}$$

Birth: We select u.a.r a node $y \in \tau_0$ and associate it to a new track.

$$q_{birth}(\cdot) = (1/|\tau_0|)p_T$$

Death: We choose u.a.r a track τ_k and delete it. The nodes belonging to the deleted track are moved to τ_0.

$$q_{death}(\cdot) = (1/K)p_T$$

Split: We u.a.r select a track τ_k and then select a break point according to the probability $br_k(i)$. The nodes in the track which are after the break point are moved to a new track.

$$q_{split}(\cdot) = (1/K)p_T br_k(i)$$

Merge: If a track's (τ_{k_1}) end node is in the parent set of another track's (τ_{k_2}) start node, this pair of two tracks is candidate for a merge move. We select u.a.r a pair of tracks from candidates and merge the two tracks into a new track $\tau_k = \{\tau_{k_1}\} \cup \{\tau_{k_2}\}$. Let C denote the number of candidates (which is also used for the remaining moves).

$$q_{merge}(\cdot) = (1/C)p_T$$

Switch: If there exist two break points p, q in two tracks τ_{k_1}, τ_{k_2}, such that $\tau_{k_1}(t_p)$ is in the parent set of $\tau_{k_2}(t_{q+1})$ and $\tau_{k_2}(t_q)$ is in the parent set of $\tau_{k_1}(t_{p+1})$ as well, this pair of nodes is one candidate for a switch move. We u.a.r select a candidate and define two new tracks as:

$$\tau'_{k_1} = \{\tau_{k_1}(t_1), ..., \tau_{k_1}(t_p), \tau_{k_2}(t_{q+1}), ..., \tau_{k_2}(t_{|\tau_{k_2}|})\}$$
$$\tau'_{k_2} = \{\tau_{k_2}(t_1), ..., \tau_{k_2}(t_q), \tau_{k_1}(t_{p+1}), ..., \tau_{k_1}(t_{|\tau_{k_1}|})\}.$$

$$q_{switch}(\cdot) = (1/C)p_T$$

4.3. Spatial Moves

Temporal moves only change the label of rectangles in the cover. However, since detected moving regions do not always correspond to a single target (they may represent parts of a target or delineate multiple targets moving closely to each other), merely using temporal moves cannot probe the spatial cover of foreground. Hence, we propose a set of spatial moves to segment, aggregate or diffuse detected regions to infer the best cover of foreground. The spatial and temporal moves are interdependent: the result of a spatial move is evaluated within temporal moves, meanwhile the result of a temporal move will guide subsequent spatial moves. The prior of segmentation and aggregation moves is p_S and the prior of diffusion moves is p_D.

Forward Segmentation: If more than one track's prediction $\bar{\tau}_k(t)$ has enough overlap with one covering rectangle at time t, the rectangle is a candidate for a segmentation move and the tracks are *related tracks* of the candidates. We randomly select such a candidate y and for each *related track* τ_k generate a new covering rectangle $\tau'_k(t)$ according to the probability $S(\tau'_k(t)|\bar{\tau}_k(t))$

$$S(\mathbf{y}'_t|\mathbf{y}_t) \sim N\left(\mathbf{y}_t + \alpha \frac{dE}{d\mathbf{x}}|_{x=\mathbf{y}_t}, \mathbf{u}\right) \quad (14)$$

where $E = -\log L_A(x|y_t)$ is the appearance energy function, α is a scalar to control the step size and $\mathbf{u}$ is a Gaussian white noise to avoid local minimum. In practice, we adopt the spatio-scale mean shift vector[4], which provides an approximation of the gradient of the appearance likelihood in terms of position and scale. The newly generated covering rectangles will take the place of $\tau_k(t)$.

$$q_{seg}(\cdot) = (1/C)p_S \prod_k S(\tau'_k(t)|\bar{\tau}_k(t))$$

If there exist partial foreground regions which are not covered by newly added rectangles, the MBRs of the uncovered foreground regions are added into τ_0. This is applied to all spatial moves.

Forward Aggregation: If one track's prediction has enough overlap with more than one covering rectangle at time t, this forms a candidate for an aggregation move. We randomly select a candidate and merge the related covering rectangles to form a new rectangle.

$$q_{agg}(\cdot) = (1/C)p_S$$

Diffusion: We randomly select one covering rectangle in a track, update its position and size to generate a new covering rectangle $\tau'_k(t)$ according the probability $S(\tau'_k(t)|\bar{\tau}_k(t))$.

$$q_{dif}(\cdot) = (1/\sum_{k=1}^{K} |\tau_k|)p_D S(\tau'_k(t)|\bar{\tau}_k(t))$$

5. Experiments

Comparative experiments on both simulation and real data are reported in this section. To evaluate the performance of our approach quantitatively, we adopt the metric "Sequence Tracking Detection Accuracy" (STDA) proposed in [12], which is a spatio-temporal based measure penalizing fragmentation in the temporal and the spatial domains. STDA produces a real number value between zero and one (worst and best possible performance, respectively).

5.1. Parameters Training

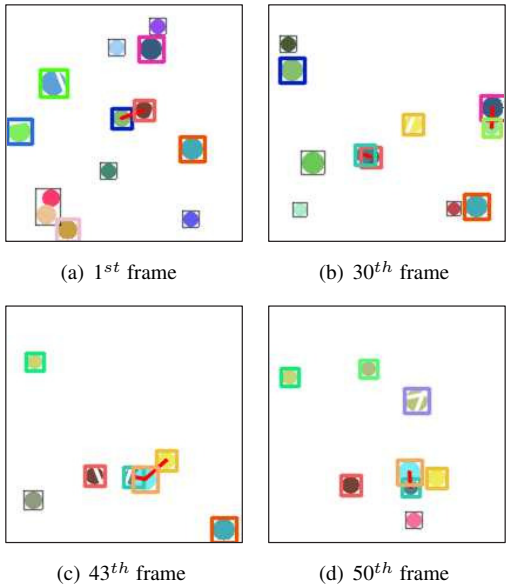
Properly selecting the parameters in Eq. 12 is necessary to assure the Markov chain converges to the correct distribution. Here, we propose an automatic solution to determine the parameters in such a probabilistic model. As mentioned in Section 4, we only need to compute the ratio of $\pi(\omega')/\pi(\omega)$ in Markov chain transition. We use this property to establish constraints on the parameters in the model. First we determine parameters in the motion model, i.e. Q and R in Eq.9. Then we start with the best cover ω^* obtained from ground truth and use the temporal and spatial moves to degrade the best cover to ω_i. For each ω_i, we have a constraint that $\pi(\omega^*)/\pi(\omega_i) \geq 1$, which provides a linear inequality in terms of the parameters. After collecting multiple constraints, we use Linear Programming to find a solution of positive parameters with a minimum sum. However, due to the ambiguity existing in the ground truth, a few conflict constraints may exist. Thus, in our experiment, 5,000 constraints, which covers most of cases of different moves from multiple sequences in one data set, are sequentially generated and added to a constraint set. Any constraint, which conflicts with the existing set, is ignored. Each LP problem is solved by GNU Linear Programming Kit (GLPK). A desired Markov chain transition and a correct MAP solution are ensured by the trained parameters.

5.2. Simulation results

To demonstrate the concept of our approach, we design simulation experiments. In a $L \times L$ square region, there are K (unknown number) moving discs. Each disc presents an independent color appearance and an independent constant velocity and scale change in the 2D region. False alarms (non-overlapping with targets) are u.a.r located in the scene and the number of false alarms is an uniform distribution on $[0, FA]$. If the number of existing targets in the square region is less than the upper bound N, a target will be added randomly. We also add several bars as occlusions in the scene. This static occlusion causes a target to break into several foreground regions. This simulates real scenarios when foreground regions are fragmented due to noisy background modelling. These sequences challenge many existing data association methods due to the frequent occlusions and fragmented observations. The input to our tracking algorithm contains merely foreground regions in each frame without using any model information. Figure 4 gives the result of our spatio-temporal MCMC data association algorithm. Colored and black rectangles display the targets and false alarms respectively. Red links indicate the spatial segmentation happens between nodes.

We compare the tolerance of the target density and false alarms with other methods, including a JPDAF based method from [6], the MHT from [5] and our own algorithm with only temporal moves. For each different setting, we generate 20 sequences and each sequence contains $T = 50$ frames. The MCMC sampler was run for a total of 10K iterations where the first 15% iterations consist solely of temporal moves. The average score from multiple runs of our method is reported. To make the comparison fair, all four methods employ the same motion and appearance likelihood. Figure 5(a) compares the performance when the number of targets increases. Figure 5(b) shows the tolerance to false alarms for different methods. Because we consider the spatial and temporal association seamlessly, our method is able to handle the case when split or merged observations exist.

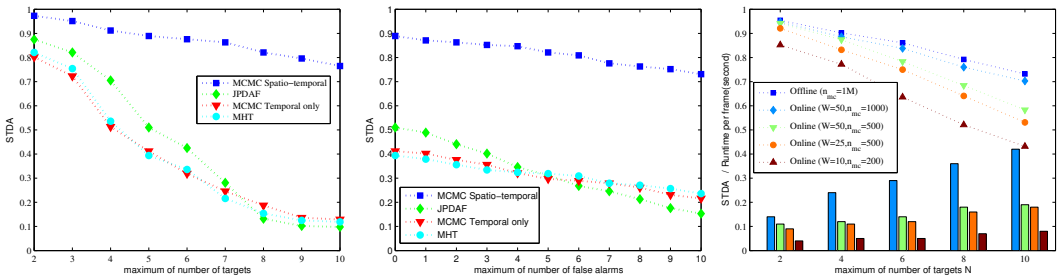
To extend our algorithm for long sequences, we implement the proposed association algorithm as an online algorithm within a sliding window containing the latest W frames. The cover of the current sliding window at time t is initialized with the best cover obtained at $t-1$. The comparison between online and offline version is shown in



(a) 1^{st} frame (b) 30^{th} frame

(c) 43^{th} frame (d) 50^{th} frame

Figure 4. Simulation result $L = 200$, $N = 7$, $FA = 7$ and $T = 50$. Color rectangles indicate the IDs of targets. Targets may split or merge when they appear.



(a) $FA = 0$ and $T = 50$ (b) $N = 5$ and $T = 50$ (c) $FA = 0$ and $T = 1000$

Figure 5. (a)STDA as the function of N the maximum number of targets, (b)STDA as the function of FA the number of false alarms, (c)STDA and runtime (second) for online/offline, different W window size and n_{mc} number of samplings.

Figure 5(c). By implementing the online version, we reduce the complexity of data association and control the delay of output for long sequences.

5.3. Real scenarios

We show results and evaluations on three video sets to demonstrate the effectiveness of our method in real scenarios. The first set is a selection from CLEAR [1], which is captured with a stationary camera, mounted a few meters above the ground and looking down towards a street. The targets in the scene include vehicles and pedestrians. The second set, called "campus ground set", is captured with a stationary camera on a tripod. The foreground in second set is clear, however the inter-target occlusion is intensive. The third set is a selection from VIVID-I and II data set, which is captured from UAV cameras. The main difficulty of the third data set comes from noisy foreground regions and false alarms caused by erroneous registration and parallax. The input to our tracking algorithm contains foreground regions which are extracted using a dynamic background model estimated within a sliding window. Tracking is performed automatically from the detected blobs without any initialization. In the case of a moving camera, the images are first stabilized using an affine motion model [6]. Table 1 gives the quantitative comparison, where the complete track is defined as 80% of the trajectory is tracked and no ID changes. In the experiments, we use online tracking with a sliding window $W = 50$ and $n_{mc} = 1000$. The tracking process runs around 3 fps on P4 3.0 GHz PC. Some foreground regions used as input and tracking results are shown in Figure 6.

	Frames	Tracks in GT	STDA		Complete tracks	
			method1	**method2**	method1	**method2**
CLEAR	17,580	152	0.539	**0.772**	96	**139**
Campusground	6,300	54	0.406	**0.846**	23	**50**
VIVID	12,690	61	0.256	**0.471**	28	**46**

Table 1. Comparative results on three real data sets. Method 1: JPDAF in[6]; Method 2: the proposed method.

One advantage of our tracking algorithm (shown in both simulation and real data set) is worth highlighting. Because the bi-directional (forward/backward) sampling is applied in a symmetric way, our approach can deal with the case where targets are merged or split when they appear. Figure 7 illustrates the comparison to the algorithms with only forward or backward inference on the image sequence in "campus ground set". The colors at the bottom of each chart correspond to labels allocated by the algorithm for the three moving persons in the sequence, while the red bars correspond to mislabeled targets due to merged observations. The proposed bi-directional sampling allows to estimate the trajectories and label them consistently throughout the sequence.

6. Conclusion and Discussion

We have presented a framework to find a global optimal spatio-temporal association which maximizes the consistency of motion and appearance of targets over time. Our method overcomes problems encountered with one-to-one mapping between observations and targets. A data driven MCMC method is used to sample the solution space efficiently and the forward and backward inferences enhance the search performance. Compared to other data association algorithms, the proposed method shows remarkable improvement both temporally (*i.e.* consistency of labels) and spatially (*i.e.* accuracy of outlined regions).

The work can be extended along the following lines: first, the target motion model can be extended to a more general model. Second, our framework can naturally incorporate object model information in two ways: 1) we can assign a model likelihood for each node to extend our likelihood function. 2) we will also use model information to drive the MCMC proposal. Third, tracking failures caused by long term occlusions can be resolved by data association at the level of tracklets.

Acknowledgements

This work was supported by MURI-ARO W911NF-06-1-0094.

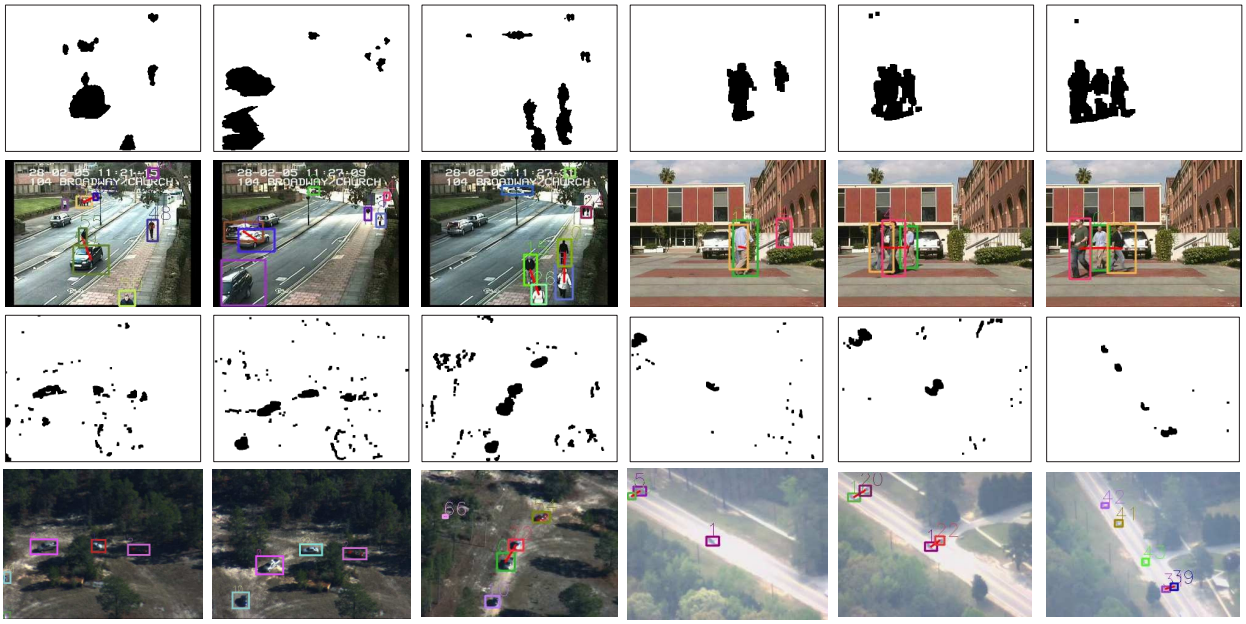
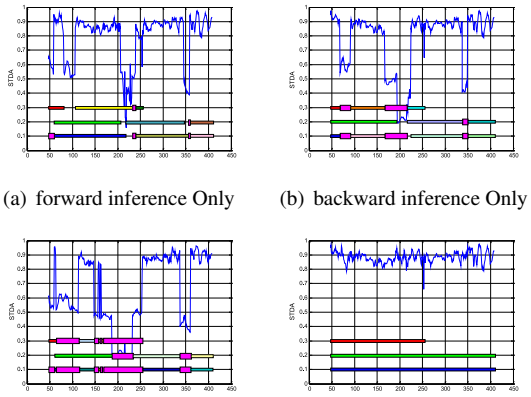


Figure 6. Experiment results of real scenarios from both stationary cameras and Unmanned Aerial Vehicle (UAV) cameras



(a) forward inference Only (b) backward inference Only

(c) JPDAF in [6] (d) Proposed method

Figure 7. The STDA and ID label for each frame

References

[1] http://www.clear-evaluation.org/.

[2] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Proc. Conf. on Information Sciences and Systems*, 1980.

[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, pages 744–750, 2006.

[4] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR*, pages 234–240, 2003.

[5] I. Cox and S. Hingorani. An efficient implementation of Reid's MHT algorithm and its evaluation for the purpose of visual tracking. In *ICPR*, pages 437–443, 1994.

[6] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *CVPR*, volume 1, pages 267–272, Jun 2003.

[7] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *PAMI*, (11):1805–1819, 2005.

[8] Z. Khan, T. Balch, and F. Dellaert. Multitarget tracking with split and merged measurements. In *CVPR*, volume 1, pages 605–610, 2005.

[9] P. S. Maybeck. *Stochastic models, estimation, and control*. Mathematics in Science and Engineering. 1979.

[10] A. Mittal and L. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, (3):189–203, 2003.

[11] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004.

[12] P. S. R. Kasturi, D. Goldgof and V. Manohar. Performance evaluation protocol for text and face detection and tracking in video analysis and content extraction. In *(VACE-II). Technical Report*.

[13] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Contr*, 24(6):84–90, Dec 1979.

[14] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *CVPR*, pages 962–969, 2005.

[15] Z. Tu and S. Zhu. Image segmentation by Data Driven Markov Chain Monte Carlo. *IEEE PAMI*, 24(5):657–674, 2002.

[16] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *CVPR*, pages 834–841, 2004.

[17] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *CVPR*, pages 406–413, 2004.