Integrated Detection and Tracking for Multiple Moving Objects using Data-Driven MCMC Data Association

Qian Yu, and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
{qianyu, medioni}@usc.edu

Abstract

We propose a framework to address the multiple target tracking problem, which is to recover trajectories of targets of interest over time from noisy observations. Due to occlusions by targets and static objects, parallax or other moving objects, foreground regions cannot represents targets faithfully although motion segmentation is usually computationally efficient. We adopt the real Adaboost classifier to generate meaningful candidate rectangles to interpret the foreground regions. Tracks are generated from these candidates according to the smoothness of motion, appearance and model likelihood overtime. To avoid enumerating all possible joint associations, we take a Data Driven Markov Chain Monte Carlo (DD-MCMC) approach which samples the solution space efficiently. The sampling is driven by an informed proposal scheme controlled by a joint probability model combining motion, appearance and model information. Comparative experiments with quantitative evaluations are provided.

1. Introduction

Tracking has become increasingly powerful in recent years, largely due to the adoption of statistical models which allow effective motion segmentation and object detection. It has been standard, however, to treat detection and tracking as separate processes. The terminology "detection" we define here refers to motion segmentation and object detection, both of which have been widely adopted as input for tracking algorithms.

Motion detection is used to segment independent moving regions, which is regarded as a computationally cheap solution, especially for stationary cameras. However, due to crowed scene, noisy background modelling, motion blobs based representation of observations, which do not faithfully correspond to targets, causes serious difficulties in tracking. Object detection is usually based on statistical model (pattern) of targets' appearance, such as shape, texture, silhouette to detect one type of target such as humans or vehicles. Pattern-based object detection is more robust to illumination change and camera motion. However, object detection (especially rotation and scale invariant object detection), which will evaluate all location (scale, rotation), is usually computational demanding for tracking purpose.

If a track is regarded as a sequence of shapes produced in the spatio-temporal space by a target, the critical constraint in tracking is the smoothness of motion, appearance and model information of such a track over time. We formulate multiple target tracking as a way to find a meaningful interpretation of foreground regions with optimal smoothness. The "meaningful" interpretation of foreground refers to the likelihood that interpreted foreground regions belong to one type of targets, such as humans or vehicles. To avoid making noisy binary detection decision, we adopt a real Adaboost classifier to assign model likelihoods for candidate detection responses, which are then associated over a sliding window according to the spatio-temporal smoothness. Due to the high computational complexity of such an association scheme, a Data-Driven Markov Chain Monte Carlo (DD-MCMC) [8] method is proposed to search for the global optimum in the solution space.

Accuracy and efficiency of detection within one frame is difficult to achieve due to the limited temporal information. In this paper, we address this limitation and lead to a framework to combine detection and tracking in a principled manner. Instead of generating binary detection results which are then pipelined into the tracking procedure, we propose to take detection as a proposal procedure and the final decision is considered according to the smoothness of motion, appearance and model information overtime. The key contribution of this paper is to propose a framework to unify multiple object detection and tracking by explicitly using spatio-temporal smoothness in motion, appearance and model information.

The rest of this paper is organized as follows. The related work is reviewed in Section 2. We formulate the multiple

target tracking problem and present our data driven MCMC data association algorithm in Section 3 and 4 respectively. We discusses how to determine the parameters used in our probabilistic model by Linear Programming and provides experimental results in Section 5, followed by conclusion and discussion in Section 6.

2. Related work

In the past decades multiple target tracking has been a very active research field. Among the large body of work, the multiple hypothesis tracker (MHT) [2] and joint probabilistic data association filter (JPDAF) [9], 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 [10]. 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). Both methods have to explicitly handle track initialization, growing and termination. Moreover, any practical implementation of these algorithms requires pruning/ merging all hypotheses to a smaller hypotheses set.

Due to the ambiguity existing at any one time instant, deferred logical inference makes 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 terms of the depth of observations. Markov Chain Monte Carlo has been successfully applied in solving the data association problem in many areas, such as structure from motion [11] and multiple object tracking [12, 13]. In [11], the authors applied MCMC to simulate the distribution of the association between the 2D image features and 3D points to avoid a hard decision of the data association. In [12], a MCMC method is applied in multiple object tracking problem to solve the data association between observations and objects at one time instant. In [13], the authors originally proposed a MCMC method to organize the temporal association between objects and punctual observations over time. This method is extended in [14] to solve the data association in spatio-temporal space where observations cannot be approximated as points. The consistency in both motion and appearance is used to find the best interpretation of the foreground region. However, due to the lack of the object model information, this method cannot deal with the crowded scenarios, such as the case that two or more objects are always merged together.

Many blob tracking systems have been developed in recent years, e.g. ([4], [5] and [15]), to track objects from foreground regions. In [15], sequential tracking methods use pairwise Markov random field (MRF) based prior to model the interaction between targets at one time instant. There are some other methods which start from foreground blobs and introduce rigid [5] or articulated [4] object model to interpret foreground regions.

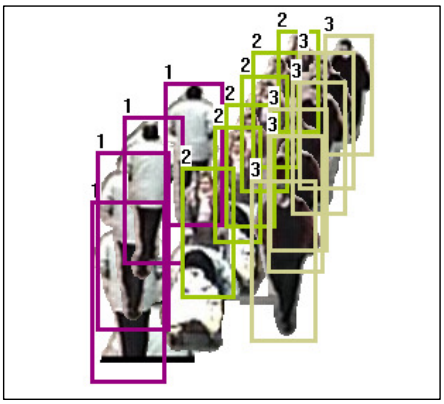


Figure 1. One possible interpretation, or say meaningful covering, of the foreground regions over time, which includes three tracks (τ_1, τ_2, τ_3). τ_0 is not shown for the clarity purpose. The frame rate of overlayed foreground region is down-sampled.

Object detection is also widely used as input for tracking system. In [16], an offline trained part-based Adaboost human detector is used to provide input for tracking, where explicit track initialization, growing and termination are applied to deal with occlusions. In [17], Adaboost detection results are originally used to propose particles, which are then input to particle filters. In [18], the Adaboost image likelihood, together with the motion and appearance likelihood, is incorporated in the posterior distribution and the MCMC is applied to find a MAP solution. However, the model information is not used in a more informed proposal distribution of the MCMC sampling.

3. Bayesian formulation

Suppose that within the time interval $[1, T]$, there is an unknown number K of targets. The "target" we refer to here can be a category of objects, for examples, vehicles, or humans. The observations come from foreground regions. Let y_t denotes the set of foreground regions at time t, Y is the set of all available foreground regions within $[1, T]$. If we regard a track as the trajectory of one target traveling in the spatio-temporal space, the tracking problem can be defined as given the observation Y, inferring of an unknown number K of tracks.

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

where τ_0 is the set of false alarms, $\tau_k, (k = 1, \ldots, K)$ is the k^{th} track. Each τ_k in ω is defined as a sequence of shapes, which forms a cover (interpretation) of foreground

regions. For simplicity, we use rectangles, which actually represent the location and scale of a target, to represent the shapes covering foreground regions. The tracking problem can be regarded as finding a meaningful interpretation (in term of one type of objects) of foreground to maximize the smoothness of motion and appearance in spatio-temporal space. Figure 1 shows one possible interpretation of foreground regions, which includes three tracks (τ_1, τ_2, τ_3).

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:

$$\omega^* = \arg\max(p(\omega|Y)) = \arg\max(p(Y|\omega)p(\omega)) \quad (2)$$

In the rest of this section, we discuss the prior and likelihood model used in our method.

3.1. Prior Probability $p(\omega)$

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. Also, one track should not have unnecessary overlap with other tracks. Accordingly, we adopt the prior probability of a cover ω as the product of the following terms.

$$p(\omega) = p(L)p(F)p(O) \quad (3)$$

Let $|\tau_k|$ denote the length, i.e. the number of elements in τ_k. The exponential model $p(L)$ of the length of each track can be represented as follows.

$$p(L) = \prod_{k=1}^{K} \frac{1}{z_0} \exp(\lambda_0 |\tau_k|) \quad (4)$$

Let F denote the size of τ_0. The prior model $p(F)$ for false alarms can be represented as follows.

$$p(F) = \frac{1}{z_1} \exp(-\lambda_1 F) \quad (5)$$

The exponential model $p(O)$ penalizes overlap between different tracks.

$$p(O) = \prod_{t=1}^{T} \frac{1}{z_2} \exp(-\lambda_2 \Gamma(t)) \quad (6)$$

where $\Gamma(t)$ denotes the average overlap ratio of different tracks at time t.

$$\Gamma(t) = \frac{\sum_{\tau_i(t)\cap\tau_j(t)\neq\emptyset} \frac{\tau_i(t)\cap\tau_j(t)}{\tau_i(t)\cup\tau_j(t)}}{|\tau_i(t)\cap\tau_j(t)\neq\emptyset|} \quad (7)$$

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

Given one cover, the likelihood of one target is independent of others. We represent the elements 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.

The joint likelihood of one possible cover, which considers motion, appearance and model likelihood can be factorized as follows.

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

where (a) in Eq.8 encodes the motion and appearance consistency within the trajectory of each target. $L_M \circ L_A$ in Eq.8 denote the motion and appearance likelihood of track τ_k. Besides motion and appearance consistency, the proper cover of foreground regions should comply with model of targets of interest. L_D represents the model likelihood of the bounding box in τ_k at time t_i.

3.2.1 Motion likelihood

For each target, we consider a linear kinematic model:

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

where $\mathbf{x}_t^k$ is the kinematic state vector, which includes the position (u, v), scale s (we assume a fixed length-width ratio) and the first order derivatives $(\dot{u}, \dot{v}, \dot{s})$ in 2D image coordinates. A and H represent the transition matrix and observation matrix respectively. Measurement $\mathbf{y}_t^k$ in Eq.9 corresponds to the position and scale 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.

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

where $\mathbf{e}_i = \tau_k(t_i) - \bar{\tau}_k(t_i)$ and $T_i = H^T\bar{P}_i H + R$. 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. Note that, if missing detection happens in τ_k at time t, or say there is not observation at time t for track k, the prior estimate is assigned to the posterior estimate.

3.2.2 Appearance Likelihood

In order to model the appearance of each detected region, we adopt the non-parametric histogram-based descriptor [7] to represent the appearance of image blobs. The appearance likelihood is defined as follows.

$$L_A(\tau_k(t_{i+1})|\tau_k(t_i)) = \frac{1}{z_3} \exp(-\lambda_3 D(\tau_k(t_i), \tau_k(t_{i+1}))) \quad (11)$$

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

3.2.3 Model likelihood

The likelihood, $L_D(\tau_k(t_i))$ in Eq. 8, encodes the probability of the image covered by $\tau_k(t_i)$ is a target. This likelihood only focuses on the pattern of one type of target and is independent of motion. We build this model likelihood model using a boosted classifier with the Edgelet features [19] for two types of objects, *i.e.* human and vehicle. Given a set of labeled patterns $\{(I_i, l_i)\}$, the customary AdaBoost procedure learns a weighted combination of base weak classifiers, $H(I) = \sum_{t=0}^{T} h_t(I)$, where $h_t(I)$ is the LUT-based weak classifier [20] chosen for the round t of boosting. The boolean decision is made by $sign(H(I) - b)$, where b is the threshold for controlling the detection and false alarm rates. For each LUT-based weak classifier, the range of the edgelet feature $f_{edgelet} \in [0, 1]$ is divided into n bins. The weak classifier can be formulated as:

$$h_t(I) = \frac{1}{2} \sum_{j=1}^{n} \ln(\frac{W_{+1}^j + \xi}{W_{-1}^j + \xi}) \delta_j(f_{edgelet}(I)) \tag{12}$$

where $W_m^j = P(f_{edgelet}(I) \in bin_j, l = m)$, δ_j is the indicator function of $f_{edgelet}(I) \in bin_j$. Instead of using the Boolean detector, we use a likelihood function by exponentiating the confidence output by the AdaBoost classifier.

$$L_D(I) = \frac{1}{z_4} \exp\left(\lambda_4 \sum\nolimits_{t=0}^{T} h_t(I)\right) \tag{13}$$

where λ_4 is a positive constant. In order to reduce the false alarm rate, we use a bootstrap procedure that iteratively adds false alarms that are reported by the trained strong classifier and then re-train the detectors using the old positive and the new extended negative sets.

3.3. Discussion of the posterior

With some manipulation, we combine the prior $p(\omega)$ in Eq. 3 and the likelihood $p(\omega|Y)$ in Eq. 8 to have the full posterior represented in Eq. 14.

$$\begin{aligned}
&p(\omega|Y) \propto \exp\{-C_0 S_{len} - C_1 K - C_2 F - C_3 S_{olp} \\
&-C_4 S_{app} - C_5 S_{mdl} - S_{mot}\} \\
&S_{len} = \left(\sum_{k=1}^{K} |\tau_k|\right), S_{olp} = \left(\sum_{T=1}^{T} \Gamma(t)\right) \\
&S_{app} = \sum_{k=1}^{K} \sum_{i=1}^{L_k - 1} D(\tau_k(t_i), \tau_k(t_{i+1})) \\
&S_{mot} = \sum_{k=1}^{K} \sum_{i=1}^{L_k} \left(\log\left(\det(T_i)^{1/2}\right) + \tfrac{1}{2} \mathbf{e}_i^T P_i^{-1} \mathbf{e}_i\right) \\
&S_{mdl} = \sum_{k=1}^{K} \sum_{i=1}^{L_k} H(\tau_k(t_i))
\end{aligned} \tag{14}$$

where $C_0, \cdots, C_5$ are positive real constants. Eq. 14 reveals that the MAP estimation is equivalent to finding the minimum of an energy function. To keep the scale of the problem in a more manageable scale and reduce the delay of decision, especially for a long sequence (*i.e.* a large T), we propose the association algorithm as an online algorithm within a fixed time duration, which is called a sliding window and contains the latest W frames. Although the problem is restricted within a sliding window, the MAP estimate of such a posterior or even determining the parameters for Eq.14 is not trivial. In the experiment section, we discuss how to determine the parameters in such a probabilistic model by Linear Programming.

4. Data-Driven MCMC Data Association

We propose to use the MCMC method to find the MAP estimate of the posterior in Eq.14. 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. Data-driven MCMC provides a principled framework to integrate top-down (hypothesis-driven) and bottom-up (data driven) models in a principled way [6]. Data-driven proposal techniques, which use the object detection and the smoothness in motion and appearance, drive the Markov chain dynamics and achieve an efficient method to find the global optimum.

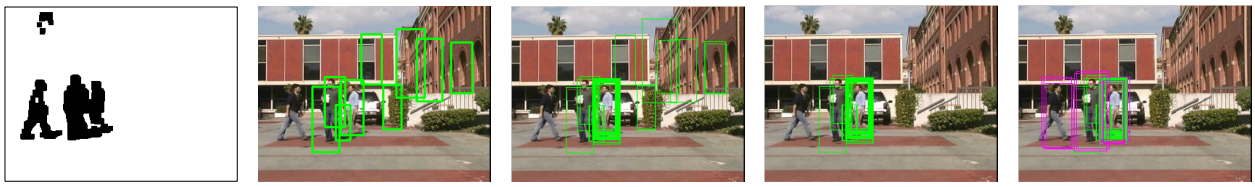
The input of the algorithm is the set of original foreground Y, initial cover ω_0 and the total number of samples n_{mc}. Each move is sampled according to its own prior probability. Note that, instead of keeping all samples, we only keep the cover with the maximum posterior since we don't need the whole distribution but the MAP estimate. And for the same reason, there is no burn-in procedure. Given the stationary distribution $\pi(\omega) = p(\omega|Y)$, the acceptance ratio at time i in the Markov chain $A(\omega_i, \omega')$ is defined as follows.

$$A(\omega_i, \omega') = \min\left(1, \frac{\pi(\omega')^{1/T_i} q(\omega|\omega')}{\pi(\omega)^{1/T_i} q(\omega'|\omega)}\right) \tag{15}$$

where T_i is temperature at time i in the Markov chain. To avoid the temperature drops too fast at the end of the Markov chain, we adopt the cool schedule $T_i = (C \ln(i + T_0))^{-1}$. In such a Markov chain's transition, there is no need to acquire the exact value of $\pi(\omega)$, but up to a constant. In each move, the computation cost for each MCMC move is actually low, since we usually change ω to ω' partially and the incremental part of $\pi(\omega')/\pi(\omega)$ is computed unless a change affects the other part of $\pi(\omega)$.

4.1. Candidate proposal

A candidate of one target is a rectangle in 2D image space which covers large enough foreground region, *i.e.* $R_{\text{cov}} = \frac{\text{covered foreground}}{\text{area of rectangle}} > Th_{\text{cov}}$. We adopt two approaches to propose candidates which are then associated according their spatio-temporal smoothness. The first one is to use the Adaboost classifier to generate candidates at different scales, whose response is strong enough *i.e.* $\sum_{t=0}^{T} h_t(I) > Th_{mld}$. The second way to generate candidates is based on motion. Given the best ω^* formed in



(a) (b) (c) (d) (e)

Figure 2. (a) foreground regions (b)binary Adaboost detection (c)all Adaboost responses (d)Adaboost responses filtered by motion (e) Combined candidates from Adaboost and motion

the previous sliding window $[t, t+W]$, the candidates for each track $\tau_k \in \omega^*$ at $t+W+1$ are generated according to prediction of τ_k with a Gaussian noisy $\mathbf{w} \sim \mathcal{N}(0, Q)$. The probability of taking Adaboost candidates is α and the probability of taking motion candidates is $(1-\alpha)$. Before the tracking starts, $\alpha = 1$, *i.e.* all the candidates are generated by the Adaboost classifier. The two types of candidates generated during tracking are shown in Figure 2

In experiments, we use the integral image of the foreground mask to compute the foreground area covered by a rectangle efficiently. We also adjust a low Th_{mld} to adopt a high detection rate. In the MCMC data association algorithm, the detected candidate rectangles are considered to form tracks according to the smoothness in both motion, appearance and model likelihood over time.

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. $L(y|\tau_k(t_i))$ 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 $q(\omega'|\omega)$.

4.2. Markov Chain Dynamics

Within the time span $[1, T]$, the "future" and "past" information is symmetric: *e.g*, we can extend a track in the positive time direction , but also in 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 describe sampling in the positive time direction and the sampling in the other direction proceeds in a symmetric way.

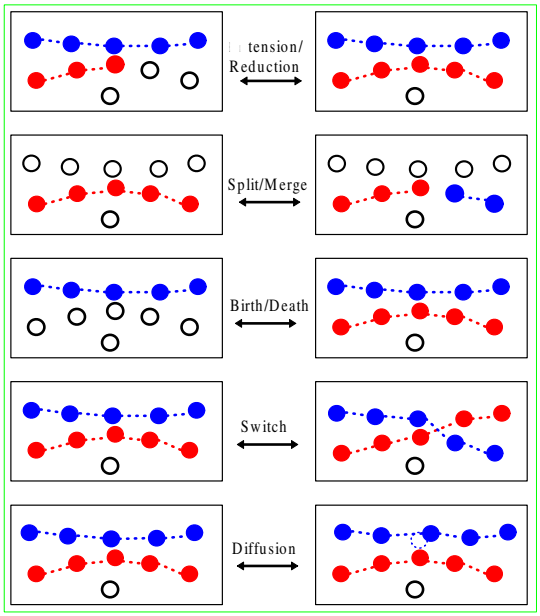


Figure 3. Illustration of MCMC moves. White circles represent false alarms.

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, where n is the number of actual extensions.

$$q_{extension}(\cdot) = \frac{1}{K} p_{extension} \prod_{i=1}^{n} \gamma \cdot ex_k(y_i)$$

$$ex_k(y_i) = \frac{(-\log L(y_i|\tau_k(t_{i-1}))+1)^{-1}}{\sum_{y_i \in child(\tau_k(t_{i-1}))} (-\log L(y_i|\tau_k(t_{i-1}))+1)^{-1}}$$

When a track is extended, the smoothness of the motion, appearance and model likelihood $ex_k(y_i)$ is applied to guide the MCMC proposal.

Reduction We u.a.r select a track k to reduce its length. We then select a break point i 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.

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

When a track reduces its length, the break point is selected according to its non-smoothness likelihood, which will prefer to keep the part with higher smoothness.

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.

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

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.

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

Birth We select a node $y_1 \in \tau_0$ according to the probability $p_D(y)$ and assign it as the first node of a new track. Starting from the first node, we then select one false alarm node from the set of child nodes according to the joint likelihood $ex_k(y)$. We keep on extending the track according to a probability $\gamma \in [0, 1]$.

$$q_{birth}(\cdot) = p_D(y_1)p_{birth}\prod_{i=1}^{n}\gamma ex_k(y_i)$$
$$p_D(y_1) = \frac{\log L_D(y_1)}{\sum_{y\in\tau_0}\log L_D(y)}$$

Here the model information is used to drive the MCMC proposal, *i.e.* we can give birth to a track only if a node is likely to be a target. A similar, but more specialized idea is used in [4], where human head position is used to add a human hypothesis.

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.

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

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

Diffusion Randomly select one track and select covering rectangle $\tau_k(t)$ according to $br_k(y)$, generate a new covering rectangle $\tau'_k(t)$ according to the probability $S(\tau'_k(t)|\bar{\tau}_k(t))$

$$S(\tau'_k(t)|\bar{\tau}_k(t)) \sim N\left(\bar{\tau}_k(t) + \alpha\frac{dE}{d\mathbf{x}}\Big|_{x=\bar{\tau}_k(t)}, \mathbf{u}\right) \qquad (16)$$

where $E = -\log L_A\left(x|\bar{\tau}_k(t)\right)$ 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 [3], 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 $\tau_k(t)$.

$$q_{dif}(\cdot) = \frac{1}{K}br_k(i)p_{dif}S(\tau'_k(t)|\bar{\tau}_k(t))$$

5. Experiments

Comparative experiments on several real data sets are reported in this section. We also submitted videos with this document to illustrate the presented experimental results.

5.1. Determining parameters

Properly selecting the parameters in Eq. 14 is required to assure the Markov chain converges to a correct distribution. Here, we propose an automatic solution to determine the parameters in such a probabilistic model. Given a possible ω, the posterior $p(\omega|Y)$ can be represented a linear equation of the parameters in Eq.14. As we mentioned in Section 4, we only need to compute the ratio of $\pi(\omega')/\pi(\omega)$ in the Markov chain transition. Note that annealing by decreasing the temperature does not affect this property.

We use this property to establish constraints on the parameters in the model. Starting with the best cover ω^* obtained from ground truth, we use the MCMC 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 in Eq.14. 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 ground truth, 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. A constraint, which conflicts with the existing set, is ignored. Each LP problem is solved by GNU Linear Programming Kit (GLPK). The final parameters computed by LP make sure the desired the Markov chain transition and a correct MAP solution.

To adapt our algorithm for long sequences, we implement the proposed association algorithm as an online algorithm within a sliding window of size W containing the latest W frames, which contains enough temporal information. The cover of the current sliding window at time t is initialized with the best cover obtained at $t-1$. By implementing the online version, we reduce the complexity of data association and control the delay of output for long sequences.

5.2. Real Scenario

We show results and evaluations on data sets of two types of objects, humans and vehicles. For each type of objects, an Adaboost classifier, which adopts edgelet features [19] is trained offline. Targets or scenes in the test data set are not included in either positive or negative samples. The input to our tracking algorithm contains, the offline trained classifier, original images and foreground regions which are extracted using a dynamic background model estimated within a sliding window [7]. To determine the parameters of the posterior, we label the ground truth of tracks in several sliding windows for each data set. The ground truth for each target at one time is labelled in the same way as we collect positive samples in training the Adaboost classifier to make sure the labelled ground truth is meaningful in terms of our trained classifiers.

We compare both spatial and temporal accuracy of our algorithm with other methods, including the JPDAF based method from [7], the MHT from [10] and our own algorithm with only binary Adaboost detection (*i.e*. clustered from Adaboost responses). To make the comparison at the same level, all four methods employ the same motion, appearance and model likelihood. The comparison is shown in Table 1. A target at one time is tracked only if the ratio of the overlapping over the union between the covering rectangle and the ground truth is larger than a threshold (we setup a spatial threshold of 0.75). A complete track is defined as 80% of the trajectory is tracked and no ID changes.

5.2.1 Side-view vehicle tracking

We test our algorithm on side-view vehicle tracking to demonstrate the effectiveness of our approach. The Adaboost classifier is trained with 1660 positive side-view (heading left) car samples with size of 75×30 pixels. The side-view vehicle test sequences are captured at a regular urban crossing with a stationary camera. The resolution of the test sequence is 640×480. For this scenario, we only focus on the vehicles with a side-view, *i.e*. the turning vehicles are ignored in ground truth. Since the classifier is trained for one orientation, during tracking, the classifier is mirrored horizontally to generate candidates in both orientations. In this experiments, we use online tracking with a sliding window $W = 30$ and the number of MCMC samples is 1000 which runs around 3 fps on P4 3.0Hz PC.

5.2.2 Human tracking

For human tracking, we adopt the full-body detector [16] trained with 1742 positive samples with size of 24×58 pixels. The samples are aligned according to the positions of head and feet. The first data set for human tracking 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 resolution of the CLEAR set is 640×480 pixels. The main difficulty of the first data set for detection is targets appear at multiple scale of size due to the perspective effect. Also, missing detection happens due to the low resolution when targets moves far away from the camera. The second set, called "campus ground set", is captured with a stationary camera on a tripod around 5 feet above the ground. The resolution of second data set is 320×240. The second set has clear foreground, however contains many inter-target occlusions. In this experiments, we use online tracking with a sliding window $W = 50$ and the number of MCMC samples is 1000, which runs 3-5 fps on P4 3.0Hz PC..

	Tracks in GT	Complete Tracks			
		JPDAF	MHT	Binary	**Our method**
Side View	73	32	39	36	**58**
CLEAR	69	45	49	47	**56**
Campusground	62	32	35	28	**49**

Table 1. Comparative results on three real data sets.

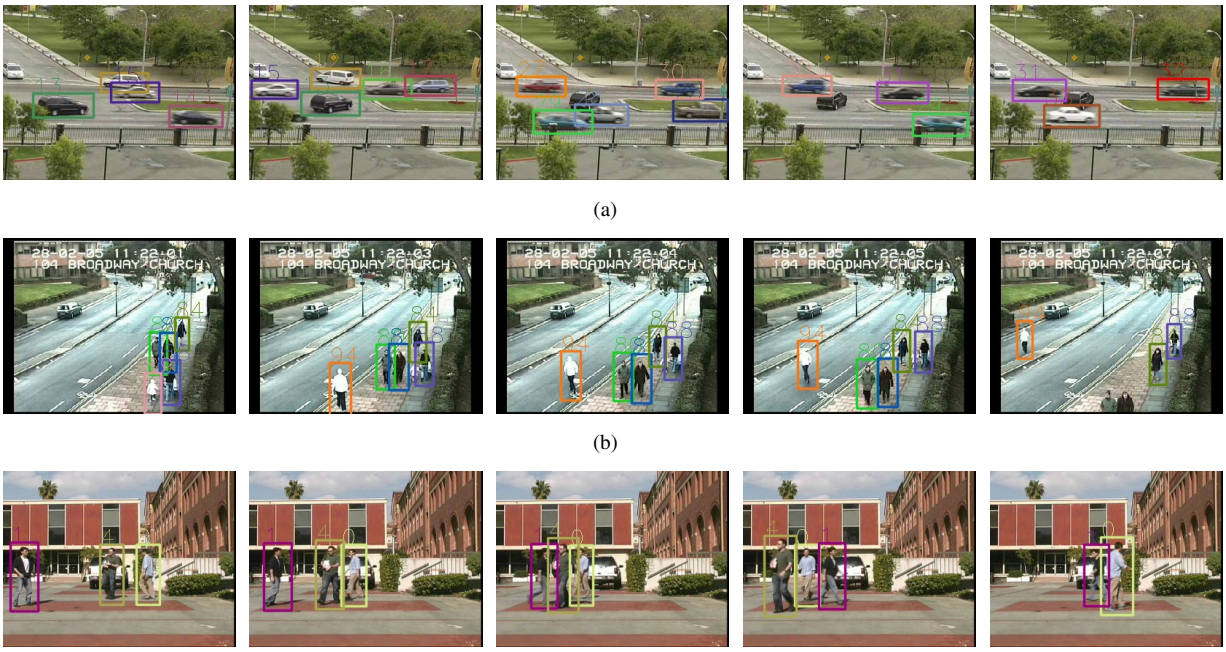
6. Conclusion and Discussion

We have presented a framework to unify multiple object detection and tracking by explicitly using spatio-temporal smoothness in motion, appearance and model information. To avoid making noisy binary detection decision, we adopt a real Adaboost classifier to assign model likelihoods for candidate detection responses, which are then associated to tracks over a sliding window according to the spatio-temporal smoothness. 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 the remarkable improvement in terms of 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 motion model can be more general than the linear motion; second, a part-based detector will definitely lead to more informed proposal distribution and thus increase the overall performance; third, tracking failures caused by long term occlusions can be resolved by data association on the level of tracklets.

Acknowledge

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



(a)

(b)

(c)

Figure 4. Sample tracking results on two types of objects. (a) is from the side-view vehicle data set (b) is from the CLEAR[1] data set (c) is from the campus-ground data set.

References

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

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

[3] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR*, page 234C240, 2003.

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

[5] X. Song and R. Nevatia. A model-based vehicle segmentation method for tracking. In *ICCV*, pages 1124–1131, 2005.

[6] Z. Tu and S. Zhu. Image segmentation by data-driven markov chain monte carlo. In *PAMI*, volume 24, pages 657–673, 2002.

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

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

[9] 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.

[10] 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.

[11] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. Structure from motion without correspondence. In *CVPR*, 2000.

[12] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *ECCV (4)*, pages 279–290, 2004.

[13] 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.

[14] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *"CVPR"*, pages 1–8, 2007.

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

[16] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006.

[17] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, pages 28–39, 2004.

[18] Q. Yu, I. Cohen, G. Medioni, and B. Wu. Boosted markov chain monte carlo data association for multiple target detection and tracking. In *ICPR*, pages 675–678, 2006.

[19] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *In ICCV*, pages IV: 90– 97, 2005.

[20] B. Wu, H. AI, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 79–85, 2004.