Trajectory Parsing by Cluster Sampling in Spatio-temporal Graph

Xiaobai Liu[1,3] Liang Lin[2,3], Song-chun Zhu[2,3] and Hai Jin[1]
[1] SCTS&CGCL, HUST, China
[2] University of California, Los Angeles
[3] Lotus Hill Research Institute
xbliu@lotushill.org, liang,sczhu@stat.ucla.edu, hjin@hust.edu.cn

Abstract

The objective of this paper is to parse object trajectories in surveillance video against occlusion, interruption, and background clutter. We present a spatio-temporal graph (ST-Graph) representation and a cluster sampling algorithm via deferred inference. An object trajectory in the ST-Graph is represented by a bundle of "motion primitives", each of which consists of a small number of matched features (interesting patches) generated by adaptive feature pursuit and a tracking process. Each motion primitive is a graph vertex and has six bonds connecting to neighboring vertices. Based on the ST-Graph, we jointly solve three tasks: 1)spatial segmentation; 2)temporal correspondence and 3)object recognition, by flipping the labels of the motion primitives. We also adapt the scene geometric and statistical information as strong prior. Then the inference computation is formulated in a Markov Chain and solved by an efficient cluster sampling. We apply the proposed approach to various challenging videos from a number of public datasets and show it outperform other state of the art methods.

1. Introduction

This paper presents a novel trajectory parsing framework to track and preserve identity of multiple moving objects in a visual surveillance application. As shown in Fig. 1, we represent each trajectory using a bundle of moving primitives in a spatio-temporal graph and aim to jointly solve three challenging tasks: (i) spatial segmentation/grouping at each frame, (ii) temporal matching/tracking, and (iii) object recognition.

In the literature, tracking algorithms mostly focus on recognition of moving objects and corresponding features. Objects can be represented by **shape**, such as points [6], structural primitives [22], silhouettes and contours [2], and skeleton models [19] etc., or **appearance**, including density probability [7], template [26], and active appearance models [23] etc. Examples of tracking features are color [18], edges [2], optical flow [23], and texture [15]. However, it is still a open problem to recover the correct correspondence under long occlusion and clutter in complex scenes.

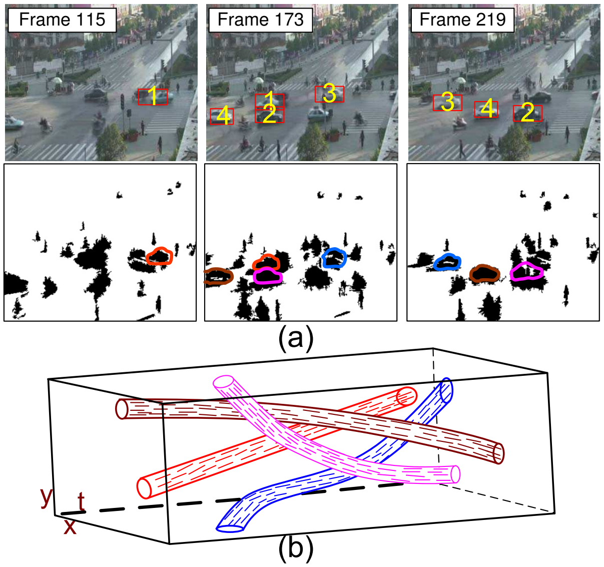

Figure 1. Trajectory parsing via deferred observations. (a) Three frames (top row) from a image sequence and corresponding foreground mask (bottom row). Object trajectories are shown in mask image and different colors denote different recognized objects. We use the background modeling component to detect moving pixels as initial proposal. (b) The parsed trajectories in a perspective view. Each trajectory consists of a bundle of moving primitives.

Applying tracking methods to trajectory analysis can be directly viewed as **(i) sequential inference** based on current observation. Representative methods are particle filters [25] and online detectors by boosting ensemble [21]. These methods often work well in punctual videos, where objects and observed moving blobs are mostly subject to one-to-one mapping in each time instance. The performance is enhanced by introducing graphical spatial prior [25] for moving objects or multi-view model [3].

(ii) Deferred inference based on a period of observation was first proposed by Reid [9]. Many deterministic searching algorithms, such as dynamic programming [10, 12], multiple hypothesis tracker [11, 20], and joint probabilistic data-association filter [5, 14] are widely used for deferred logical inference. However, in real visual surveillance, a moving blob or region cannot be treated as an object faithfully, due to inaccurate segmentations caused by occlusion, conglutination, or spurious motion. One moving object can decomposes as several foreground blobs and several objects may be conglutinated together. Therefore, the large solution space entail simultaneously spatial segmentation and temporal tracking using stochastic inference with efficient driven features, as the pioneer work using Data-driven MCMC (DDMCMC) in tracking by [22, 16].

The closet approach to ours is presented by Yu et al [16], which also uses stochastic MCMC algorithm for spatio-temporal association. However, there are four significant differences. (1) Their approach inferred trajectory based on foreground blobs that are not reliable in complex scenes, whereas we introduce the motion primitives (see Fig. 2) to overcome scene perturbation and reduce the solution dimensions in video sequence. (2) They performed stochastic sampling for temporal and spatial association independently in two iterative MCMC dynamics, in contrast, we use a more efficient cluster sampling in a spatio-temporal graph to jointly solve the segmentation (spatial) and tracking (temporal) together; (3) We additionally explore scene context information as strong prior for trajectory parsing. (4) We integrate object recognition with inference.

We introduce our framework in following three aspects: spatio-temporal graph representation, scene context modeling, and cluster sampling inference.

(i). Spatio-temporal graph is constructed based on the deferred observations. We first pursue and match the sequential small features (interesting patches) to generate a number of cubic cells (in 3D coordinate), called “motion primitives”, as shown in Fig. 2. Using these primitives as vertices with 6 bounds connecting to the neighbors, the spatio-temporal graph is defined and the trajectory parsing is simply formulated as a graph labeling (coloring) problem.

(ii). Cluster sampling is designed to efficiently explore the joint space of spatio-temporal graph coloring. The move between two solution states is a reversible jump following the Metropolis-Hastings method, containing two steps: 1) generating a cluster of motion primitives as a connected component after turning off some edges probabilistically, and 2) flipping the label (color) of the cluster as a whole. The jointly computation of segmentation and tracking essentially integrates the spatial appearance and temporal similarity of moving object, and make them boost each other for fast convergence.

(iii). Scene context information [8] provides strong prior for trajectory parsing inference in spatio-temporal graphs. We model two types of context information as important cues. (1) Statistical path model, that consists of a set of reference trajectories clustered from training data in supervised way, provides global motion prior for tracking objects. (2) Surface property and camera geometry parameters, can be further use to predict object location and size for each category. The density and recognition label of moving objects are also statistically learned as weak prior.

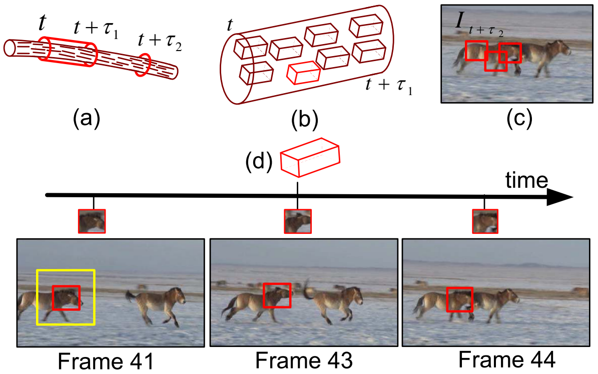

Figure 2. Illustration of a trajectory with motion primitives. A typical trajectory is shown in (a); (b) shows a cropped trajectory from t to $t+\tau_1$; (c) shows a cross section of the trajectory at time $t+\tau_2$ and 3 interesting features/patches (in red box); (d)Each motion primitive is a series of matched path/features in a short time span, represented as a “cubic cell” in $3D$ coordinates. The features are selected by the similarity discrepancy with respect to the local surrounding background region (denoted by the yellow box).

The remainder of this paper is arranged as follows. We first present the spatio-temporal graph representation in Sect. 2, and formulate the problem in Bayesian framework in Sect. 3. We introduce the inference algorithm in Sect. 4 and demonstrate the experiments in Sect. 5. We conclude this paper Section 6 with a summary.

2. Spatio-temporal graph representation

Given an observed image sequence $I_{[0,\tau]} = (I_T, I_{T+1}, \ldots, I_{T+\tau})$, we first compute the foreground map $\wedge_{t,F}$ and background map $\wedge_{t,B}$, using the background modeling algorithm proposed in [24]. A few small feature patches are then selected from $\wedge_{t,F}$ and matched to the following frames. We group a short sequence of matched features from adjacent $(3 \sim 5)$ frames and define each group as one **motion primitive**. The motion primitives are defined in the $3D$ coordinate for dimension reduction (like the super-pixel in 2D image segmentation). Fig. 2 intuitively illustrate a trajectory and the motion primitives in perspective coordinates.

In order to generate the motion primitives, we introduce a feature pursuit scheme and a template matching algorithm. Let $B_i = \{B_{i,t}; i = 0, \ldots, N, t = 0, \ldots, \tau\}$ denote the ith object model over video sequence $I_{[0,\tau]}$ and N denote the object (trajectory) number. We can further define the feature template $B_{i,t}$ as,

$$B_{i,t} = \{B_{i,t,j}, j = 1, \ldots K_{i,t}\} \tag{1}$$

where $K_{i,t}$ is the number of the features selected for the ith object at time t. As illustrated in Fig. 3(b), $B_{0,t}$ denotes the remaining feature patches at time t. Each feature $B_{i,t,j}$ is defined as

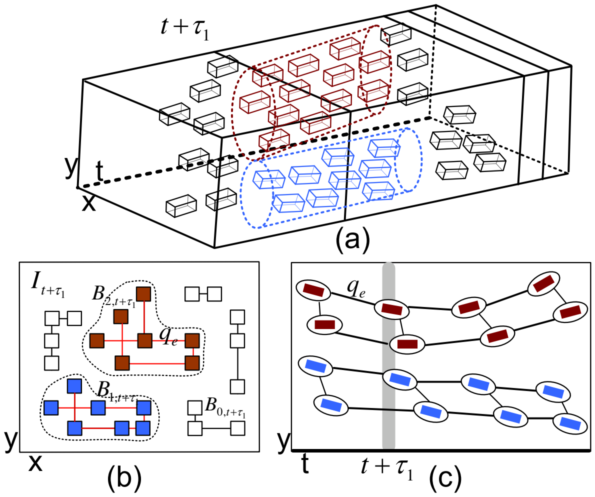

Figure 3. (a) Spatio-temporal graph, in which each vertex is a motion primitive and the edge probability q_e is defined on local similarity. Each vertex has 6 bonds connecting to neighboring vertices. The trajectory parsing is formulated as graph coloring problem. Each cylinder (in $3D$ coordinates (x, y, t)) with different colors denotes a cluster of motion primitives and we can sample the cluster as a whole in computation step; (b-c) $2D$ view of the clusters projected on (x, y) and (y, t) coordinates.

$$B_{i,t,j} = \{x_{i,t,j}, y_{i,t,j}, w_{i,t,j}, h_{i,t,j}, F_{i,t,j}\} \quad (2)$$

where $(x_{i,t,j}, y_{i,t,j})$ and $(w_{i,t,j}, h_{i,t,j})$ are center position and size of the window covered by feature $B_{i,t,j}$. $F_{i,t,j} = h(B_{i,t,j})$ is the feature descriptor extracted from the patch $\wedge_{B_{i,t,j}}$, and we use a 12 bins normalized histogram of gradient in this work. Letting $P_i(\wedge_{i,t,F}|B_{i,t})$ denote the foreground distribution of the ith object and $q(\wedge_{f,G})$ denote the reference distribution, we can pursue $B_{i,t,j}$ from a overcomplete dictionary by maximizing the likelihood ratio of foreground distribution with respect to the background distribution,

$$\frac{P_i(\wedge_{i,t,F}|B_t)}{q(\wedge_F)} = \prod_j \frac{P_i(\wedge_{B_{i,t,j}}|B_{i,t,j})}{q(\wedge_{B_{i,t,j}})}. \quad (3)$$

The background distribution $q(\wedge_{B_{i,t,j}})$ is collected from the surrounding region of the feature patch $\wedge_{B_{i,t,j}}$, as illustrated in Fig. 2(d). Here we reasonably assume the feature being conditionally independent with each other given the object model at frame I_t. This model will maximize the contrast of the foreground and background by sequentially selecting the most discriminative feature set, similar to the shared sketch algorithm proposed in [26].

We match each selected feature into successive frames to obtain the matching correspondence $\Psi_{i,t}$ at time t,

$$\Psi_{i,t}\colon \{B_{i,t,1}, B_{i,t,2}, \ldots\} \cup \{\phi\} \to \{B_{i,t+1,1}, B_{i,t+1,2}, \ldots\} \cup \{\phi\}$$

which can be optimized by,

$$\begin{aligned}\Psi^*_{i,t} &= \arg\min_{\Psi_{i,t}} \sum_{j=0}^{K_{i,t}} D^B(B_{i,t,j}, \Psi_{i,t}(B_{i,t,j})) \quad (4)\\ &+\lambda_0 \sum_{j=0}^{K_{i,t}} \mathbf{1}(\Psi_{i,t}(B_{i,t,j}) = \phi)\end{aligned}$$

$$D^B(B_{j_1}, B_{j_2}) = \{KL(h(B_{j_1})||h(B_{j_2}))\} \quad (5)$$

Where λ_0 is a penalty factor for unmatched features, $\mathbf{1}(\cdot) \in \{0, 1\}$ is an indicator function for a Boolean variable, and $D^B(\cdot)$ returns the KL divergence of two feature distributions. Note that the features at time t may be mapped to/from $\emptyset$ due to object moving, lighting change, or occlusion.

Thus, we represent each trajectory using a set of motion primitives. Letting C_i denote the ith trajectory and $\mathcal{P}_{i,j}$ denote the jth motion primitive of the ith trajectory, we have,

$$C_i = \{\mathcal{P}_{i,j} = \{x_{i,j}, y_{i,j}, \{B_{i,j,t}\}\}\} \quad (6)$$

where $i = 0, \ldots, N, j = 0, \ldots, N_i, t \in [t_{i,j,b}, t_{i,j,d}]$. N denotes the total number of trajectories, N_i is the primitive number of the ith trajectory and $(x_{i,j}, y_{i,j})$ denotes the central position. $[t_{i,j,b}, t_{i,j,d}]$ is the lifespan of the primitive $\mathcal{P}_{i,j}$, e.g 3 $\sim$ 5 frames.

A **spatio-temporal graph** $G = (V, E)$ is thus constructed with the motion primitives being graph vertices V. we define a vertex as $v_j = \mathcal{P}_j, j \in [1, N_V]$ with N_V being the total number of vertices. In order to form a sparse adjacent graph structure, we assume each vertex have 6 bonds connecting to neighbors and G is thus a neighborhood system in $3D$ coordinates (x, y, t). Fig. 3 illustrates the ST-Graph as well as its projection on coordinates (x, y) and (y, t). In this ST-Graph, a cluster is one connected component of primitives (called a "**ST-CCP**") and will receive the same label in sampling inference (described in Sect. 4).

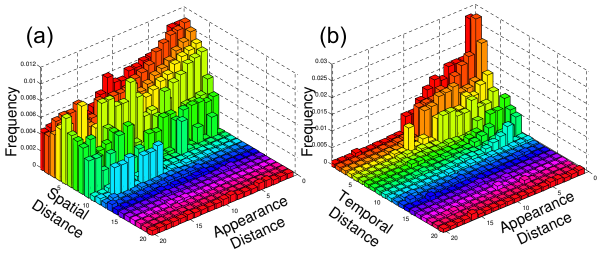

Figure 4. Similarity probability histograms learned from training data. (a)The joint frequency of appearance distance and spatial distance. (b)The joint frequency of appearance distance and temporal distance(sequential frames).

For each edge $e =< u, v >\in E, V_u = \mathcal{P}_u, V_v = \mathcal{P}_v \in V$, we introduce an auxiliary variable q_e which indicates how likely the two connected vertices belong to the same trajectory, i.e. receiving the same label. We compute q_e by combining two types of measurements: geometric distance and appearance distance,

$$q_e = H^S(\Delta^S_{u,v}, D^P_{u,v}) \cdot H^T(\Delta^T_{u,v}, D^P_{u,v}) \qquad (7)$$

$$\begin{aligned}\Delta^S_{u,v} &= ||(x_u - x_v)||^2 + ||(y_u - y_v)||^2 && (8)\\ \Delta^T_{u,v} &= ||t_{u,b} - t_{v,b}|| && (9)\\ D^P_{u,v} &= \sum_{t_1} \min_{t_2} D^B(B_{u,t_1}, B_{v,t_2}), && (10)\\ & \quad t_1 \in [t_{u,b}, t_{v,d}], t_2 \in [t_{u,b}, t_{v,d}]\end{aligned}$$

H^S and H^T respectively denote the spatial and temporal edge probability histogram, which can be counted from a set of manually annotated trajectories. Fig. 4 shows the two probability histograms corresponding to the scene in Fig. 1(a) and it indicates the primitives share more similar appearance in temporal tracking than in spatial segmentation, such that the formed ST-CCPs are always string-like.

Compared with the independent spatial and temporal representation in [16], ST-CCP with motion primitives is able to capture both the spatial and the temporal information. Thus the trajectory parsing is equal to flipping the label of a generated ST-CCP.

3. Bayesian Formulation

Given the ST-Graph via the observed image sequence $I_{[0,\tau]} = (I_T, I_{T+1}, \ldots, I_{T+\tau})$, we define the following solution representation W as

$$W_{[0,\tau]} = \{N, C_{[0,N]}, L_{[0,N]}\} \qquad (11)$$

$$C_i = \{K_i, V_i, t_{i,b}, t_{i,d}, \Gamma_i\} \qquad (12)$$

where C_i denotes the ith object trajectory, N is the total trajectory number, $L_i \in \{'car', 'pedestrian', 'bike', 'motorcycle'\}$ is the recognition label. Each trajectory C_i with life span $[t_{i,b} t_{i,d}]$ includes K_i vertices (motion primitives) V_i and skeleton shape Γ_i . We use C_0 to collect the remaining foreground blobs and false alarm in current state. This solution essentially integrates three tasks: moving object segmentation, temporal tracking and object identity preserving.

Given the trajectories $C_{[0,N]}$, the moving object segmentation in each frame t is also defined as

$$\Pi_{[0,\tau]} = \{\pi_t = \{R_{i,t}\}\} \qquad (13)$$

$$R_{i,t} = C_{i,t} = \{x_{i,t}, y_{i,t}, w_{i,t}, h_{i,t}\} \qquad (14)$$

where $t = 1, \ldots, \tau, i = 1, \ldots, N_t$. $\Pi_{[0,\tau]}$ denotes the spatial segmentation of all trajectories at time $t \in [0, \tau]$, $R_{i,t}$ denotes the foreground region(transverse plane) of the ith trajectory at time t, defined by object position $(x_{i,t}, y_{i,t})$ and size $(w_{i,t}, h_{i,t})$. N_t is the trajectory number at time t. Note that each transverse plane $R_{i,t}$ is the foreground region covered by the feature template $B_{i,t}$ (see Eqn. 1).

We can thus solve the problem of trajectory parsing by maximizing a posterior (MAP) probability in the framework of Bayesian,

$$\begin{aligned}W^*_{[0,\tau]} &= \arg\max_W P(W_{[0,\tau]} | I_{[0,\tau]}, T) && (15)\\ &= \arg\max_W P(I_{[0,\tau]} | W_{[0,\tau]}, T; \beta) P(W_{[0,\tau]} | T; \theta)\end{aligned}$$

where T is current system time(e.g A.M.8.00), β and θ are the parameters for the likelihood and prior models.

3.1. Prior model

We define the prior model of solution $W_{[0,\tau]}$ given the system time T,

$$P(W_{[0,\tau]} | T; \theta) = P(N|T) P(L_{[0,N]} | T) P(C_{[0,N]} | T) \qquad (16)$$

$$P(C_{[0,N]} | T) = P(\Pi_{[0,\tau]} | T) \prod_{i=1}^{N} P(C_i | T)$$

where $P(\Pi_{[0,\tau]} | T)$ denotes the spatial prior of trajectories, including object location and size. $P(C_i | T)$ denotes the temporal motion consistency for each trajectory, including trajectory birth/death position, length, and global shape. $P(N|T)$ and $P(L_{[0,N]} | T)$ are trajectory density and recognition label distribution in the surveillance scene.

I. Trajectory density prior It is defined by a histogram on object number over system time T and accounts for how busy the scene is at a given time, as,

$$P(N|T) = Hist_o(N|T) \qquad (17)$$

This distribution can be directly learned from the labeled training data directly.

II. Recognition prior We utilize a multi-nomial distribution (dirichlet model) in Statistics for recognition prior over system time T. It can be learned from an initial uniform histogram, given a batch of observations.

$$P(L_{[0,N]} | T) = \prod_{i=0}^{N} P(\ell_i | T) \qquad (18)$$

$$P(\ell_i | T) = Hist_r(T) = \alpha(T) = (\alpha_1, \ldots, \alpha_4) \qquad (19)$$

where $(\alpha_1, \ldots, \alpha_4)$ denotes the histograms of 4 categories.

III. Trajectory spatial prior We assume the moving object is segmented independently at each frame t, the segmentation prior distribution is defined as

$$P(\Pi_{[0,\tau]} | T, L_{[0,N]}) = \prod_{i=0}^{N_t} \prod_{t=0}^{\tau} P(R_{i,t} | L_i) \qquad (20)$$

Unlike traditional segemtation/grouping method using Potts model [1], each type of interested objects in surveillance system, e.g. human, vehicle and bicycle, have strong prior about their physical size at each position where they may occur. For example, a pedestrian cannot be off the ground without the other support surfaces. With camera calibration and ground-plane estimation, we can calculate the expected physical size of each foreground blob in the image-plane.

Therefore, we predict object location and size according to the scene surface property and camera calibration as,

$$\begin{aligned} P(R_{i,t}|L_i) \quad &= \quad P(x_{i,t}, y_{i,t}|L_i; \theta_S) \\ & \quad\quad P(h_{i,t}, w_{i,t}|x_{i,t}, y_{i,t}, L_i; \theta_S, \theta_C) \end{aligned} \tag{21}$$

The first term denotes the prior distribution of object position, and it can be set as uniform function or counted from training videos. The second term is the marginal distribution of object size given object position in image and the recognition label. θ_S is the scene surface property model (such as ground, road and vertical planes) and θ_C is camera parametric model respectively. The implementation for these two term can be referred in the literature [8].

VI. Trajectory temporal prior We define the tracking prior following two aspects: i) object birth/deadth position and ii) global trajectory shape. Assuming these two terms are independent with each other, we have,

$$P(C_i|T) = P(t_{i,b}, t_{i,d}|T; \theta_T)P(\Gamma_i|T; \theta_T) \tag{22}$$

where the first term denotes the prior distribution of trajectory lifespan based on the birth/death map as shown in Fig. 5 (b), and the second term denotes the prior distribution of trajectory skeleton based on the path model $\Re$, which consists of a set of reference trajectories, as shown in Fig. 5 (c). θ_T denotes the parameters of the birth/death map and the path model, which can be learned from the training videos, as the work proposed by Wang et al [27]. As illustrated in Fig. 5 (d), $P(\Gamma_i|T; \theta_T)$ can be further factorized by a mixture model plus robust statistic, as

$$P(\Gamma_i|T; \theta_T) \propto \sum_{C_j \in \Re} \alpha_{i,j} e^{-\mathcal{K}(H(\Gamma_i, \Gamma_{C_j})/h_w)} + \epsilon \tag{23}$$

where $\alpha_{i,j}$ denotes the geometric distance between two trajectories, $\mathcal{K}$ is the Gaussian function with kernel size h_w, $H(\cdot)$ returns the similarity distance and ϵ is the tuning parameter for robustness. Here we calculate $H(\cdot)$ using the schema proposed by Lin [13].

3.2. Likelihood model

Given an observed video, the proposed method explains each frame I_t into three parts: (i) regions of segmented (foreground) object, (ii) false alarm foreground regions, and (iii) background regions. Formally, we have,

$$\wedge_t = \wedge_{t,F} \cup \wedge_{t,0} \cup \wedge_{t,B}, k\wedge_{t,F} = \bigcup_{i=1}^{N} R_{i,t} \tag{24}$$

Therefore, the likelihood model can be calculated from three aspects: (i) false alarm regions $\wedge_{t,0}$ and background regions $\wedge_{t,B}$ should fit the background modeling component in the surveillance system;(ii) moving objects in sequential frames should match in appearance similarity, and (iii) recognition confidence generated by a learnt detector.

$$\begin{aligned} P(I_{[0,\tau]}|W_{[0,\tau]}; \beta_B) &= \prod_{t=0}^{\tau} P(\wedge_{(t,B)}; \beta_B) P(\wedge_{t,0}; \beta_B) \\ &\cdot \prod_{i=1}^{N} \prod_{t=t_{i,b}}^{t_{i,d-1}} P(R_{i,t+1}|R_{i,t}) P(R_{i,t}|\ell_i; \beta_{Rec})) \end{aligned} \tag{25}$$

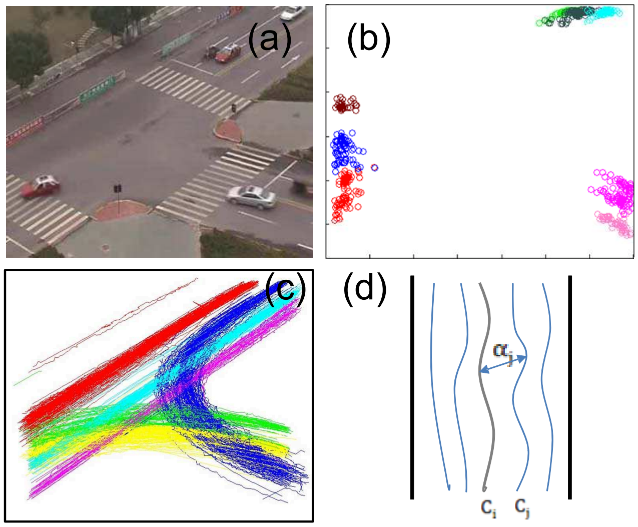

Figure 5. Scene path model for trajectory temporal prior. (a) Observed scene;(b)Trajectory birth and death position map fitted with Gaussian distribution; (c) Reference trajectories clustered from training video; (d)Trajectory skeleton. The prior distribution of trajectory skeleton can be calculated by shape matching with the reference trajectories.

where β_B denotes the parameters of the background modeling component (more details can be referred in [24]) and β_{Rec} denotes the parameters of the recognition module. The recognition module β_{Rec} embedded in our framework is proposed by [15] and it can be replaced by any other state-of-art method. The correspondence similarity $P(R_{i,t+1}|I_{R_{i,t}})$ can be further factorized as

$$P(R_{i,t+1}|R_{i,t}; \beta) \propto e^{-D^R(R_{i,t}, R_{i,t+1})} \tag{26}$$

where $D^R(\cdot)$ is the distance metrics of two matched regions, and here we use $D^B(\cdot)$ defined in Eqn. 4 for instead. Note we neglect the system time T here for clarity.

4. Inference

Given the spatio-temporal graph $G =< V, E >$ via deferred observations $[0, \tau]$, the trajectory parsing can be formulated as graph multi-coloring problem in $3D$ coordinate. As shown in Fig. 3, we find the solution is essentially a joint form of foreground object segmentation and matching. Note the recognition $L_{[0,\tau]}$ is solved with partition deterministically. In order to search for the global optimal solution in the large and complex space, we present a stochastic cluster sampling algorithm.

Our method simulates a Markov chain which visits a sequence of states in the joint solution space over time span τ, and realizes a set of reversible jumps between any two successive states. For each stochastic jump step, whether a new state is accepted is decided by the Metropolis-Hastings method which is able to guarantee the global convergence of the inference algorithm. Given two successive states A and B, the acceptance rate is defined as,

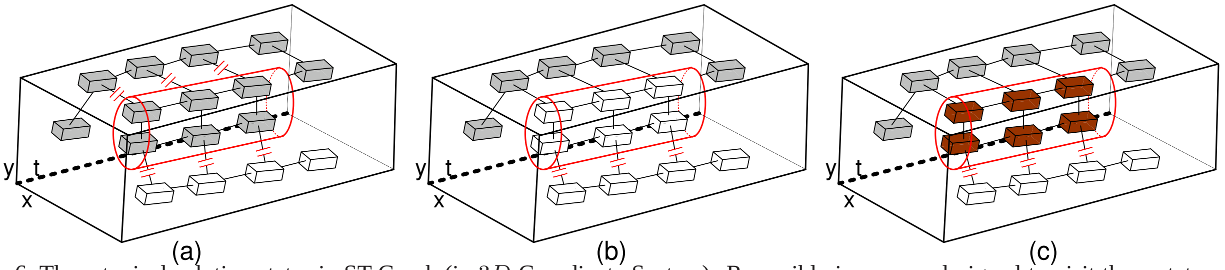

Figure 6. Three typical solution states in ST-Graph (in $3D$ Coordinate System). Reversible jumps are designed to visit these states. The red || denotes a "cut" operation to turn off the edge probabilistically. The red cylinder is a generated ST-CCP.

$$\alpha(A \to B) = \min\left(1, \frac{Q(B \to A)P(B)}{Q(A \to B)P(A)}\right) \tag{27}$$

where $P(A)$ and $P(B)$ are the posterior probability defined in Eqn. 16. $Q(B \to A)$ and $Q(A \to B)$ are proposal probability of "jumping" between two states. Following proof in [1], proposal probability ratio can be simplified in cluster sampling, which contains two steps: 1) Generating a connected cluster by sampling the edge connection; 2) Flipping the label of selected cluster. Here we perform two cluster sampler in two dynamics. It is worth mentioning that the proposed cluster sampling is an extension of the Swendsen-Wang cuts sampling [1] on the spatio-temporal representation.

Cluster generation in ST-Graph $G =< V, E >$ is to form a ST-CCP (Spatio-temporal connected component of motion primitives), by sampling the edge probability(defined in Eqn. 7). We first remove the edges that connect two different colors deterministically, and then "cut" (turn off) some edges with probability $1 - q_e$. The remaining edges form a few of clusters, ST-CCPs, denoted as the red cylinder in Fig. 6. Vertices in one ST-CCP usually share similar appearance and thus most likely belong to the same trajectory, in sense that they receive the same color.

Reversible jumps are designed to travel the states in solution space, by flipping the color of the selected ST-CCP (we u.a.r select one if there are more than one) to drive reversible jumps. There are three possible moves as shown in Fig. 6:

- **Split-and-merge.** The selected cluster is assigned to a existing color, such that a portion of a trajectory is regrouped into another existing trajectory and the total trajectory number remains N. The move between the state (a) and (c) is example.
- **Split.** The selected cluster is assigned to a new label, that is, a new trajectory is created, like the move from state (a) or (b) to state (c) in Fig. 6.
- **Merge.** A whole object is selected as a cluster and merged into another trajectory, as from state (c) to state (a) or (b) in Fig. 6.

The benefit of the cluster sampling [1] lies in the fact that we can easily compute the proposal probability ratio as,

$$\frac{Q(B \to A)}{Q(A \to B)} = \frac{q(CCP|B)}{q(CCP|A)} = \frac{\sum_{e \in \mathrm{C_B}}(1 - q_e)}{\sum_{e \in \mathrm{C_A}}(1 - q_e)} \tag{28}$$

where C_A and C_B are the "cut" around cluster, denoted as the "red ||" in Fig. 6. The edge probability is defined in Eqn. 7. Note once the ST-CCP is selected, the jump is performed uniformly.

5. Experiments

We integrate the proposed framework into a surveillance system (the detail is referred from [17]), which also include a background modeling module [24] and an object recognition module [15]. The system is capable of processing 10 - 15 frames per second on a PC with Core Duo 2.8 GHZ CPU and 4GB memory.

We first briefly introduce the system implementation and the parameters of our algorithm. In initial stage, we select 1000-3000 manually labeled frames to train the scene model as discussed in Sect. 3 using an interactive toolkit (the detail is referred from [24]). In working stage, each motion primitive is generated by feature pursuit and tracking process, with the fixed spatial size of 12×12 pixels and the adaptive temporal length of $3 \sim 5$ frames. The span of the deferred observation is set as $\tau = 30$ frames, and the observed window is moving with a step-size of 3 frames. For each window, we set the upper-bound of sampling iterations as 80.

Table 1. The average pixel-level and object-level accuracy. We compare the method proposed with other three state-of-art approaches, including the Joint Probability Data association [5], the MCMC-based particle filtering [25] and the MCMC-based Data association [16].

	Pixel-level				Object-level			
Data.	[5]	[25]	[16]	Ours	[5]	[25]	[16]	Ours
LHI	0.48	0.71	0.51	0.86	0.55	0.75	0.63	0.85
PETs	0.46	0.77	0.56	0.91	0.61	0.69	0.45	0.81
I-80	0.51	0.80	0.63	0.89	0.62	0.78	0.55	0.83

In experiments, we evaluate the system performance on three aspects: (i) multiple trajectories parsing, (ii) moving object recognition, and (iii) efficiency analysis.

Experiment I. We first evaluate the trajectory parsing using pixel- and object- level accuracy. The dataset we use contains 10 challenging scenes selected from the three datasets: LHI [4], PETs, and I-80. The criteria of bench-

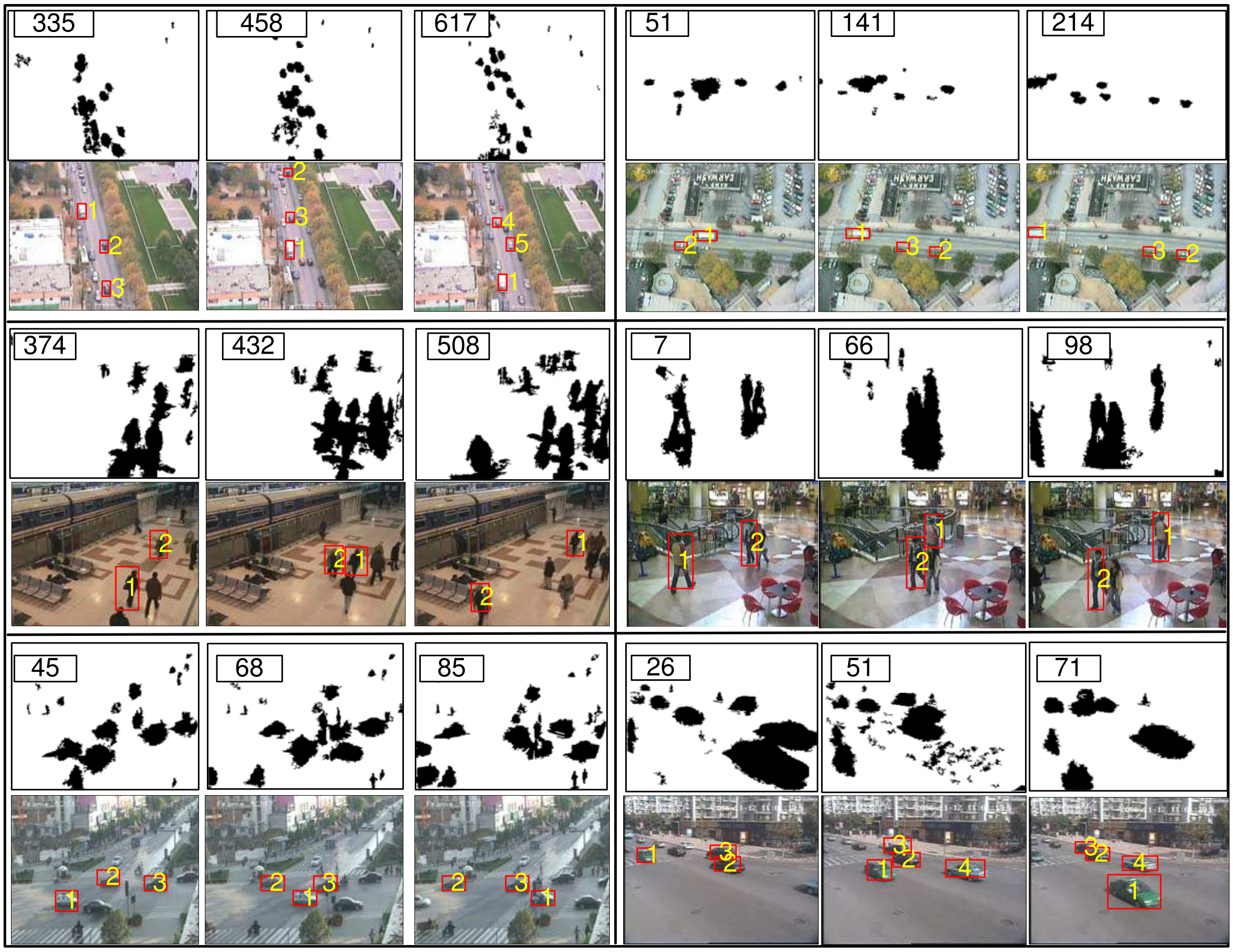

Figure 7. Some representative results on challenging scenes. Each result shows 3 images and their foreground mask. Each recovered trajectory can be identified by the bounding box (red line) with the numbers in images.

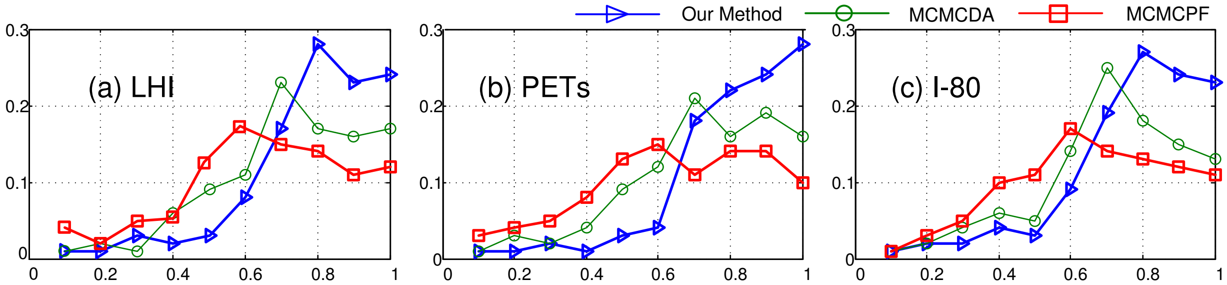

Figure 8. The frequency of the trajectory coverage rate which is defined as the ratio of traced trajectory length with respect to the trajectory time span in ground truth. Horizontal axis—coverage rate; Vertical axis—frequency. The blue, green, and red curves denote the result of our algorithm, MCMCDA [16], and MCMCPF [25] respectively.

mark includes two folds: (i) pixel-level accuracy, defined as the ratio of the foreground areas tracked correctly and object region of ground truth in each frame; (ii) object-level accuracy is defined as the ratio of correctly traced frames and the total trajectory length in ground truth. One object at each frame is counted only if the pixel-level accuracy is above 0.5. The quantitative results with comparison are reported in Tab. 1, and each row shows the result on the different datasets. We also show 6 representative results in Fig. 7, in which each cell includes the foreground region proposals by background modeling module (top row) and the results of tracking parsing(bottom row).

In addition, we introduce a novel benchmark, the average coverage rate, to demonstrate the advantages of our algorithm. The coverage rate is calculated as the ratio of traced trajectory length with respect to the trajectory time span in ground truth. Fig. 8 illustrates the frequency of average coverage rate on PETs dataset. The blue curves denote our algorithm, and the green and red curves denote the method proposed in [16] and in [25].

Experiment II. The performance of objects identity preservation in our framework is also tested on LHI dataset [4]. We plot the ROC curves of object recognition for three categories: pedestrian, sedan, and bicycle in Fig. 9. Here we adopt the the recognition algorithm proposed in [15]. The solid curves represent the recognition performance with our framework, and the dashed ones represent the result output by executing recognition independently without our framework.

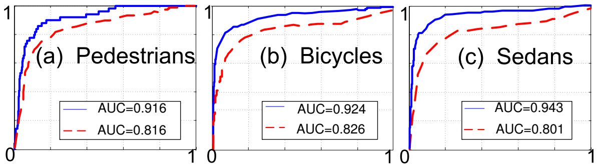

Figure 9. ROC curves of moving object recognition on the categories: (a) pedestrian, (b)bicycle, and (c) sedan. The solid curves represent the recognition with trajectory parsing and the dashed curves being executed independently without our framework. The recognition method is proposed in [15].

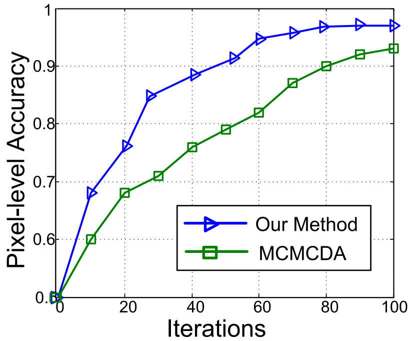

Figure 10. The average pixel-level accuracy with iteration number increasing for each move in Markov chain.

Experiment III. Algorithm computation efficiency is finally evaluated. Fig. 10 illustrates the average pixel-level accuracy increases along with adding iterations number of the simulation of Markov chain. Here we use the same data in Experiments I. Compared with the algorithm MCMCDA in [16], which performs Gibbs sampling in two MCMC dynamics, we find the cluster sampling output higher accuracy and faster convergence. There are two important reasons: 1) the cluster sampling integrates the segmentation (or grouping) and tracking (or matching) in one single move to search the solution space; 2) the edge "cut" around the cluster can be viewed as strong bottom-up proposal to drive the search.

6. Summary

In this paper, we present a novel approach to parse object trajectories from surveillance videos. Our method is distinguished from previous works by: 1) we introduce a spatio-temporal graph with vertex being motion primitive for trajectory representation; 2) we propose a flexible framework which integrates the tasks of spatial segmentation, temporal matching and object recognition in a joint solution solved by an efficiently cluster sampling algorithm and 3) we adapt variety of scene context information as the strong prior for inference. Experiments with comparisons over several challenging datasets show it outperform the state of art methods.

7. Acknowledgement

This work is done at the Lotus Hill Institute and is supported by NSFC(Grant No.60728203 and No.60832004) and China 863 program(Grant No.2008AA01Z126). The authors would like to thank Dr. Haifeng Gong and Yibiao Zhao for their helpful discussions. We acknowledge the Lotus Hill Dataset for training and testing [4].

References

[1] A. Barbu *et al*. Generalizing Swendsen-Wang for Image Analysis. *Journal of Computational and Graphical Statistics*, 2007.

[2] A. Yilmaz *et al*. Contour Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. *TPAMI*,2006

[3] A. Mittal *et al*. M2Tracker: A Multi-view Approach to Segmentating and Tracking People in A Cluttered Scene. *ECCV*, 2002.

[4] Z.Y. Yao, X. Yang, and S.C. Zhu. Introduction to A Large Scale General Purpose Groundtruth Dataset: Methodology, Annotation Tool, and Benchmarks. *EMMCVPR*, 2007.

[5] C. Rasmussen *et al*. Joint Probabilistic Techniques for Tracking Multi-part Objects. *CVPR*, 1998.

[6] C. Veenman *et al*. Resolving Motion Correspondence for Densely Moving Points. *TPAMI*, 2001.

[7] D. Comaniciu *et al*. Mean Shift: A Robust Approach Toward Feature Space Analysis. *TPAMI*, 2002.

[8] D. Hoiem *et al*. Putting Objects in Perspective. *CVPR*, 2006.

[9] D. Reid. An algorithm for Tracking Multiple Targets. *TAC*, 24(6): 84-90, 1979.

[10] I.N. Junejo *et al*. Trajectory Rectification and Path Modeling for Video Surveillance. *CVPR*, 2007.

[11] I. Cox *et al*. An efficient implementation of Reid's MHT Algorithm and its evaluation for the purpose of visual tracking. In *ICPR*, 1994.

[12] J. Berclaz *et al*. Robust people tracking with global trajectory optimization. *CVPR*, 2006.

[13] L. Lin *et al*. Layered Graph Match with Graph Editing. *CVPR*, 2007.

[14] Li Zhang *et al*. Global Data Association for Multi-Object Tracking Using Network Flows. *CVPR*, 2008.

[15] N. Dalal *et al*. Human Detection Using Oriented Histograms of Flow and Appearance. *ECCV*, 2006.

[16] Q. Yu *et al*. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. *TPAMI*, (to appear), 2008.

[17] R. Collins *et al*. A System for Video Surveillance and Monitoring. Technical Report, CMU-RI-TR-00-12, Robotics Institute, CMU, 2000.

[18] R.Collins *et al*. On-Line Selection of Discriminative Tracking Features. *TPAMI*, 2005.

[19] R. Shen *et al*. Tracking Shape Change Using A 3D Skeleton Hierarchy. *SIGGRAPH*, 2006.

[20] S.M. Khan *et al*. A Multi-view Approach to Tracking People in Dense Crowded Scenes using a planar Homography Constraint. *ECCV*, 2006.

[21] S. Avidan. Ensemble Tracking. *TPAMI*, 2007.

[22] T. Zhao *et al*. Segmentation and Tracking of Multiple Humans in Crowded Environments. *TPAMI*, 2008.

[23] T. Cootes *et al*. Robust Real-time Periodic Motion Detection, Analysis, and Applications. *TPAMI*, 2001.

[24] W. Hu *et al*. An Integrated Background Model for Video Surveillance Based on Primal Sketch and 3D Scene Geometry. *CVPR*, 2008.

[25] Z. Khan *et al*. MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets. *TPAMI*, 2005.

[26] Y.N Wu *et al*. Deformable template as Active Basis. *ICCV*, 2007.

[27] X. Wang *et al*. Learning Semantic Scene Models by Trajectory Analysis. *ECCV*, Vol.3: 110-123, 2006.