Trajectory Parsing by Cluster Sampling in Spatio-temporal Graph

Xiaobai Liu\textsuperscript{1,3}, Liang Lin\textsuperscript{2,3}, Song-Chun Zhu\textsuperscript{2,3}, Hai Jin\textsuperscript{1}

\textsuperscript{1} School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
\textsuperscript{2} Dept. of Statistic and Computer Science, University of California, Los Angeles, USA
\textsuperscript{3} Lotus Hill Research Institute, Ezhou, China

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.

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 ob-
ject faithfully, due to inaccurate segmentations caused by occlusion, conglutination, or spurious motion. One moving object can decompose 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 sequential small features (interesting patches) to generate a deferred observations. We first pursue and match the se-

(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.

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 in Sect. 6 with a summary.

2. Spatio-temporal graph representation

Given an observed image sequence \( I[0, \tau] \) = \( \{I_F, I_{I+1}, \ldots, I_{T+1}\} \), we first compute the foreground map \( \Lambda_{I,F} \) and background map \( \Lambda_{I,B} \), using the background modeling algorithm proposed in [24]. A few small feature patches are then selected from \( \Lambda_{I,F} \) and matched to the following frames. We group a short sequence of matched features from adjacent (3 ~ 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 \( i \)-th 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,

\[
B_{i,t} = \{B_{i,t,j}; j = 1, \ldots, K_{i,t}\}
\]

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

![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) maps a 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).](image)
Each cylinder (in terms as a whole in computation step; (b-c) \( \otimes \) obtain the matching correspondence sketch algorithm proposed in [26].

In [26], we pursue the most discriminative feature set, similar to the shared distribution of the foreground and background by sequentially selecting primitives and the edge probability.

The background distribution, foreground distribution with respect to the background distribution,

\[ \text{background distribution} \]

\[ \text{foreground distribution} \]

From a viewpoint of the clusters \( D \) targeted at the frame \( t \) of the object and appearance distance, (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 \( e \in E \), 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.
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

\[
W_{[0,\tau]} = \{N, C_{[0,N]}, L_{[0,N]}\}
\]

where \( C_i \) denotes the ith object trajectory, \( N \) is the total trajectory number, \( L_i \in \{\text{car'}, \text{pedestrian'}, \text{bike'}, \text{motorcycle'}\} \) 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 \( \Gamma_i \). We use \( C_i \) 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

\[
\Pi_{[0,\tau]} = \{\pi_t = \{R_{i,t}\}\}
\]

where \( t = 1, \ldots, \tau \), \( i = 1, \ldots, N_t \). \( R_{i,t} \) denotes the spatial segmentation of all trajectories at time \( t \in [0, \tau] \). \( R_{i,t} \) denotes the foreground region (transverse plane) of the \( i \)th trajectory at time \( t \), defined by object position \( (x_{i,t}, y_{i,t}, w_{i,t}, h_{i,t}) \) 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,

\[
W_{[0,\tau]}^* = \arg \max W P(W_{[0,\tau]}|I_{[0,\tau]}, T)
\]

\[
= \arg \max W P(I_{[0,\tau]}|W_{[0,\tau]}, T; \beta) P(W_{[0,\tau]}|T; \theta)
\]

where \( T \) is current system time (e.g. A.M.8.00), \( \beta \) and \( \theta \) 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 \( T \),

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

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

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,

\[
P(N|T) = Hist_{\tau}(N|T)
\]

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,

\[
P(L_{[0,N]}|T) = \prod_{i=0}^{N} P(\ell_{i}|T)
\]

\[
P(\ell_{i}|T) = Hist_{\tau}(T) = \alpha(T) = (\alpha_1, \ldots, \alpha_4)
\]

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

\[
P(\Pi_{[0,\tau]}|T, L_{[0,N]} = \prod_{i=1}^{N_t} \prod_{t=0}^{\tau} P(R_{i,t}|L_{i})
\]
where 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. \( \theta_S \) is the scene surface property model (such as ground, road and vertical planes) and \( \theta_C \) 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/death 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 \( \mathcal{R} \), which consists of a set of reference trajectories, as shown in Fig. 5 (c). \( \theta_T \) 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

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

where \( \alpha_{i,j} \) denotes the geometric distance between two trajectories, \( K \) is the Gaussian function with kernel size \( h_w \), \( H(\cdot) \) returns the similarity distance and \( \epsilon \) 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,

\[
\Lambda_t = \Lambda_{t,F} \cup \Lambda_{t,0} \cup \Lambda_{t,B}, \quad k \wedge t, F = \bigcup_{i=1}^{N} R_{t,i}
\tag{24}
\]

Therefore, the likelihood model can be calculated from three aspects: (i) false alarm regions \( \Lambda_{t,0} \) and background regions \( \Lambda_{t,B} \) 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.

\[
P(I_{[0,\tau]}|W_{[0,\tau]}; \beta_B) = \prod_{i=0}^{\tau} P(\Lambda(\Lambda_t; \beta_B)P(\Lambda_{t,0}; \beta_B)
\cdot \prod_{i=1}^{N} \prod_{t=t_i,B}^{t_{i,d-1}} P(R_{t_{i+1}}|R_{t_i}; \beta) P(R_{t_i}; \ell_i; \beta_{Rec}))
\tag{25}
\]

\[
P(R_{t_{i+1}}|R_{t_i}) = \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{26}
\]

\[
P(R_{t_{i+1}}; \ell_i; \beta_{Rec}) = \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

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 \( \beta_B \) denotes the parameters of the background modeling component (more details can be referred in [24]) and \( \beta_{Rec} \) denotes the parameters of the recognition module. The recognition module \( \beta_{Rec} \) embedded in our framework is proposed by [15] and it can be replaced by any other state-of-art method. The correspondence similarity \( P(R_{t_{i+1}}|R_{t_i}; \beta) \) can be further factorized as

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{26}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

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

4. Inference

Given the spatio-temporal graph \( G = \langle V, E \rangle \) 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 \( \tau \), 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,

\[
\alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{26}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]

\[
\sum_{C_j \in \mathcal{R}} \alpha_{i,j} e^{-D^R(R_{t_i}, R_{t_{i+1}})}
\tag{27}
\]
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.

\[
\alpha(A \rightarrow B) = \min \left(1, \frac{Q(B \rightarrow A)P(B)}{Q(A \rightarrow B)P(A)} \right) \tag{27}
\]

where \(P(A)\) and \(P(B)\) are the posterior probability defined in Eqn. 16. \(Q(B \rightarrow A)\) and \(Q(A \rightarrow B)\) 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 re-grouped 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,

\[
\frac{Q(B \rightarrow A)q(CCP|B)}{Q(A \rightarrow B)q(CCP|A)} \cdot \frac{\sum_{e \in C_B}(1 - q_e)}{\sum_{e \in C_A}(1 - q_e)} \tag{28}
\]

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 \(12 \times 12\) 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>
<thead>
<tr>
<th>Data</th>
<th>Pixel-level</th>
<th>Object-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHI</td>
<td>0.48</td>
<td>0.71</td>
</tr>
<tr>
<td>PETs</td>
<td>0.46</td>
<td>0.77</td>
</tr>
<tr>
<td>I-80</td>
<td>0.51</td>
<td>0.80</td>
</tr>
</tbody>
</table>

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

**Experiment 1.** 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 al-
Er et al. introduced 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

[4] Z.Y. Yao, X. Yang, and S.C. Zhu. Introduction to A Large Scale General Pur-

[14] Li Zhang et al. Global Data Association for Multi-Object Tracking Using Net-
[19] R. Shen et al. Tracking Shape Change Using A 3D Skeleton Hierarchy. SIG-
GRAPH, 2006.