Learning a Scene Contextual Model for Tracking and Abnormality Detection

Benjamin Yao[1] and Liang Wang[2]
[1]Department of Statistics
University of California, Los Angeles
{zyyao,sczhu}@stat.ucla.edu

Song-chun Zhu[1,2]
[2]Lotus Hill Institute for Computer Vision and
Information Science, Ezhou, 436000, China
wangliang@jdl.ac.cn

Abstract

In this paper we present a novel framework for learning contextual motion model involving multiple objects in far-field surveillance video and apply the learned model to improving the performance of objects tracking and abnormal event detection. We represent trajectory of multiple objects by a 3D graph $\mathcal{G}$ in x,y,t, which is augmented by a number of spatio-temporal relations (links) between moving and static objects in the scene (e.g. relation between crosswalk, pedestrian and car). An inhomogeneous Markov model p is defined over $\mathcal{G}$, whose parameters are estimated by MLE method and relations are pursued by a minimax entropy principle (as in texture modeling) [16] so that we can synthesize entirely new video sequences that reproduce the observed statistics from training video. With the learned model, we define the abnormality of a subgraph given its neighborhood by log-likelihood ratio test, which is estimated by importance sampling. The learned model is applied to tracking and abnormal event detection. Our experiments show that the learned model improve tracking performance and detect sophisticated abnormal events like traffic rule violation.

1. Introduction

In this paper, we present a novel approach for learning contextual motion model involving multiple objects in video surveillance. We represent objects in space and time in a Trajectory Graph $\mathcal{G}$, and augment it with a number of spatio-temporal relations that link objects (nodes) in $\mathcal{G}$. The relations are grouped into four categories (see the "Relation Library" in Figure1): i) Relation between moving objects and semantic regions in a scene (i.e. source, sink, path)[13]; ii) Relation between several frames of a single object; iii) Relation between multiple objects in one frame; iv) Relation accounting for interactions between multiple objects in a time period (e.g. car and pedestrian moving across a crosswalk at the same time). An inhomogeneous Markov model p is then defined over $\mathcal{G}$, whose parameters and relations are learned following an analysis-by-synthesis scheme. Given a set of relation estimating parameters and synthesizing samples using a Gibbs sampler until samples reproduce observed statistics over the selected relation set. Relations are pursued iteratively following a minimax in-

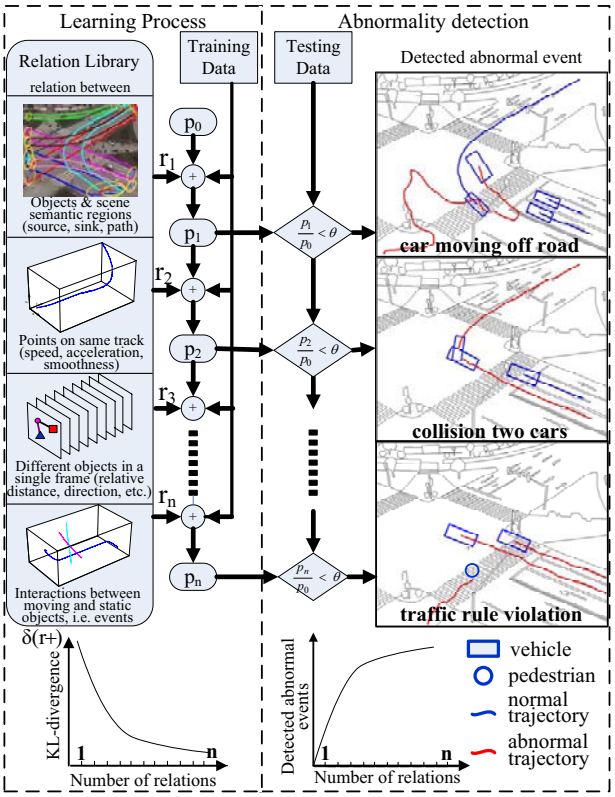

Figure 1. Diagram of algorithm. Left panel is the learning processes: At each stage, a new relation with largest information gain $\delta(r^+)$ (i.e. minimizing the KL divergence between the learned model with observed video) is pursued from a **Relation Library**. Right panel is the abnormality detection process. More sophisticated abnormal behaviors detected as new relations are added. Three typical images of detected abnormal events are shown. The bottom panel plots the information gain and abnormality recall rate along with the relation pursuit iterations.

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

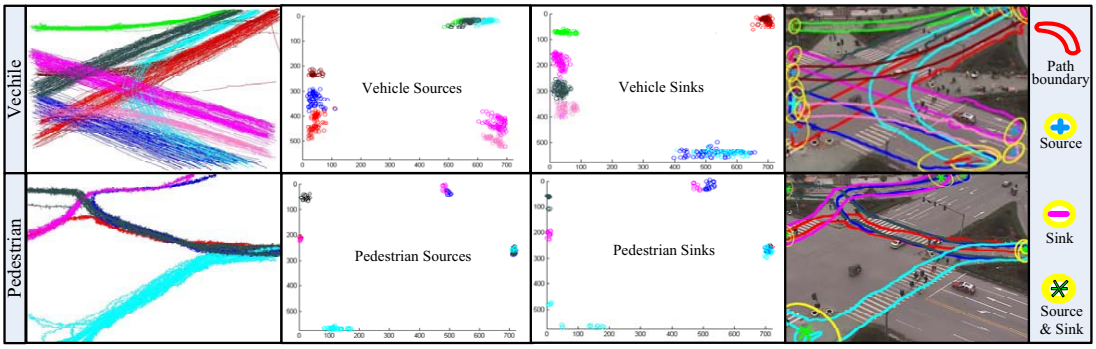

Figure 2. Source, sinks and paths. First row for vehicle, second row for pedestrian. Starting from left, the 1st figure is trajectories plotted with different colors representing different clusters of traffic. The 2nd and 3rd figure show the source and sink maps respectively, which are represented by 2D Gaussian distributions. The 4th figure illustrates the boundary of paths.To cancel perspective effects, all trajectories are projected to the bird-eye view by homographic transformation [6]

formation criteria, which sequentially selects new relations minimizing the KL divergence between learned model with observed. As shown in Figure1, at the first stage, relation between objects and scene semantic regions is pursued and thus the model is capable of detecting abnormal trajectories that move off the paths. Then with more complex and subtle relations pursued at each stages, new abnormal events with longer time periods, more structures and involving more objects are detected. This framework is closely related with a previous work for texture modeling known as the FRAME model [16] learned by minimax entropy principle. In this work, we extend the FRAME model to learning inhomogeneous Markov model in $\mathcal{G}$, which has a dynamic graphical configuration instead of a fixed neighborhood. In section 4, we test the learned model as dynamic prior of a particle filtering tracking algorithm. Moreover, we present a novel abnormality measure definition in section 5. The abnormality of a sub-graph g given its neighborhood ∂g is defined as a log-likelihood ratio test between g and expectation of learned model, which is estimated by importance sampling. Finally, we demonstrate that the algorithm is able to detect abnormal events such as object moving off road, several objects hit each other and traffic rule violation with the learned whole scene contextual motion model.

Toward the objective of modeling object event in a scene, previous work mainly falls into three categories. One school models the geometric, topological and semantic structures of the scene by clustering features of trajectories over many objects. Wang and Grimson [13] proposed a clustering method based on spatial feature of trajectory together with object features to learn semantic regions (i.e. sources, sinks and paths). Such methods discard the temporal information of trajectories, and thus fail to model interactive event between multiple objects. The second school models the object event as a sequence of states and their translations. Stauffer and Grimson [12] extracted feature prototypes from tracked objects and classified activities by hierarchically clustering the prototypes using the co-occurrence statistics of the prototypes within a track. Abnormal behaviors are detected by measuring the deviation from the learned prototype density and the co-occurrence statistics. In [11], the author proposed a Propagation Networks based representation of the events, which integrates both temporal and logic order relationships of the events. Nguyen, et al.[7] utilized the Hierarchical Hidden Markov Model to characterize the hierarchic and shared structure of complex events. Recently, Hakeem et al.[3] proposed a method to learn the dependencies between sub-events and cluster the detected sub-events in novel videos by N-cut algorithm according to the pre-learned dependencies. Another school uses a exemplar based approach to model the events [10, 15, 4]. A common character is that they define the pre-observations as exemplars and try to compose the new-come video by exemplars to determine their abnormality. For example, Irani et al. decompose existing observations into regions as the exemplars. Zhong et al.[15] use the pre-observed video clip features as prototypes. And Jiang et al.[4] model each observed trajectory by a 5 state HMM and cluster them as prototype models. To deal with variance in events, Chellappa et al.[2] use a star diagram representation to model events. The star diagram consists of several epitomes and are segmented based on linear kinematic assumption. Another interesting work worth mention here is [9] by Rosario et al., they proposed a synthetic system that mimic human activities, which simulate our work. But they employ no statistic model or learning mechanism, which is the major strength of our work.

2. Representation and Formulation

2.1. Sources, Sinks and Paths

For far-field traffic surveillance scene, semantic structure refers to source(entry), sink(exit) area and paths of vehicle and pedestrian. Given a training video sequence with tra-

jectories of objects labeled manually, semantic regions can be learned from the object trajectories by a clustering technique proposed by Wang et al. [13]. For example, in Figure 2, there are 8 paths of vehicles and 5 paths of pedestrians marked with different color. To cancel the perspective effects of slanting view, we adopt a $3D$ calibration technique based on ground plan assumption [6], which enables us to project the scene to a bird-eye-view by holographic transformation.

Considering a traffic surveillance video clip denoted by $I[0,\tau]$ as an image sequence on a 2D lattice $\mathbf{\Lambda}$ in a discrete time interval $[0,\tau] = \{0,1,\ldots,\tau\}$, let $I^{\text{obs}}[0,\tau]$ be an observed video sequence. We denote the state of an object in the sequence at time t by $\pi(t)$. We denote object state with $\pi = (X, B)$, where $X = (x, y)$ is the spatial coordinates of the object, B denotes its bounding box. Commonly, an object moves from a source to a sink following a path. Let $C[t^b, t^e]$ be the trajectory of an object: $C[t^b, t^e] = (c, l, \{\pi(t)|t = t^b, t^{b+1}, \ldots, t^e\})$, where $[t^b, t^e] \subset [0,\tau]$, t^b and t^e represent for the object's birth and death frame respectively, c denotes the type of the object (i.e. car, bus or pedestrian) and l denotes the path the object is following. The birth event is governed by a probability:

$$P_B(C) = P_B(t^b, c, l)P_B(X|c, l) = P_B(t^b, c, l)\mathcal{N}(X; \Theta(c, l))$$

where birth point is characterized with a $2D$ Gaussian distribution and $P_B(t^b, c, l)$ can be represented in a non-parametric form using Parzen windows. [14]. Similarly, the death event of a given trajectory is governed by a Gaussian distribution.

2.2. The Graph Representation for Trajectories

Assume there are K trajectories in a video sequence, we define the points of the ith trajectory at frame t as graph nodes: $V = \{v_i(t) : i = 1, 2, \ldots, K; t \in [t_i^b, t_i^e]\}$. A number of spatial, temporal and functional relations is defined between the nodes in V to form a graph with colored edges where the color indexes the type of relations.

Definition 1 A Trajectory Graph $\boldsymbol{\mathcal{G}}$ consists of a set of nodes and a number of relations $\mathcal{R}$:

$$\boldsymbol{\mathcal{G}} = \langle V, \mathcal{R} \rangle$$

The node set $V = \{v_i(t) : i = 1, 2, \ldots, K; t \in [t_i^b, t_i^e]\}$, where $v = (\pi, c, l)$, K is the number of trajectories. The relations $\mathcal{R} = \{r^1, r^2, ..., r^{N(\mathcal{R})}\}$ represents a set of directed or undirected links among a subset of nodes (sub-graph) $g \subset V$. Each relation is defined as a function $r^k = \psi^k(g_k)$ of each node's attributes.

An example of relation library is illustrated in Figure 1, which has been divided into four categories:

$\mathcal{R}_s$: *Relation between moving objects and semantic regions in a scene*. Defined over nodes from a single trajectory. $\mathcal{R}_s = \{r^{\text{dist}}\}$, represents for the distance from a trajectory to the "central line" of its corresponding path using a modified Hausdorff distance [5]. The central line is the most visited track of a path, computed by averaging all trajectories.

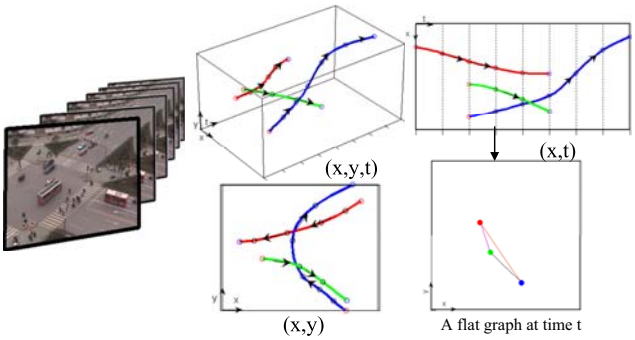

Figure 3. Illustration of a trajectory graph. Left: observed image sequence; Middle upper: a graphic view of the trajectory graph in (x,y,t) coordinate. Trajectories are represented by colored lines, Graph nodes are points in the trajectories. Middle lower: projection onto (x,y) plane, which is the common view of object track. Lower right is a slice of the trajectory graph, which denotes spatial relations between objects in a frame.

$\mathcal{R}_c$: *Relation between several frames of a single object*. Defined over nodes from a single trajectory. $\mathcal{R}_c = \{r^{\text{speed}}, r^{\text{acc}}, r^{\text{smth}}\}$. r^{speed} represents a histogram of the speed of a object at each frame. r^{acc} represents by a histogram of object accelerations at each frame. r^{smth} represents for histogram recording the absolute value of direction angle change at each frame.

$\mathcal{R}_f$: *Pair-wise relation between objects in same frame*. Defined over nodes in a frame. A flat graph is dynamically computed to represent for the spatial relationships between objects. Whether two nodes are linked or not is decided by their relative distance. (See Figure3) The $\mathcal{R}_f$ is computed over linked nodes. $\mathcal{R}_f = \{r_{ped}^{ped}, r_{veh}^{ped}, r_{veh}^{veh}\}$. For pedestrian, we use one histogram for relative distances. For vehicle, we use $n = C_3^2$ histograms representing for combination of relative directions (e.g. back-to-front, etc.).

$\mathcal{R}_e$: *Relation accounting for interaction between multiple objects in a time period*. Defined over nodes form n trajectories ($n \geqslant 1$). We only exploit one type of $\mathcal{R}_e$ in this paper: r^{event}, which is a $\mathbf{1}(\cdot)$ function measuring whether there are events that pedestrians and cars are moving across a crosswalk at the same time period. Crosswalks are labeled from the training video.(This definition can be easily extended to similar events such as car tailgating and so on)

2.3. Stochastic Models on the Trajectory Graph

We define a probability model p on the $\boldsymbol{\mathcal{G}}$. It combines both the birth/death events that affects the topological structure of the graph model and the MRF relations between

graph nodes that represent the object interactions.

We use a probabilistic model in Gibbs form to integrate the birth/death events of a $\boldsymbol{\mathcal{G}}$and the Markov relations between graph node:

$$p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R}) = \frac{1}{Z(\beta)}\exp\{-\varepsilon(\boldsymbol{\mathcal{G}})\} \tag{1}$$

where $\varepsilon(\boldsymbol{\mathcal{G}})$ is the total energy,

$$\varepsilon(\boldsymbol{\mathcal{G}}) = \sum_{i=1}^{K}\beta_i(\omega(C_i)) + \sum_{r^k\in\mathcal{R}(\boldsymbol{\mathcal{G}})}\beta_k(\psi^k(g_k)) \tag{2}$$

The model is specified by a number of parameters β and the relations set $\mathcal{R}$. The first term defines the topological events of the $\boldsymbol{\mathcal{G}}$ (i.e. birth/death events of objects). In a not-too-busy scene, the object's birth/death event can be assumed to be independent with each other. Therefore, the first term can be written as a summation. $\beta_i()$ is a function accounts for the parameters of $P_B()$ and $P_D()$ of each trajectory $\omega(C_i)$. (Since $P_B()$ is represented non-parametrically by Parzen windows, thus $\beta_i()$ is a vectorized weighting function corresponding to each bin of the histogram). The second term are typical Markov relation energy defines over the all relations in a trajectory graph. It models the spatial, temporal and event-level constraints between trajectory graph nodes.

This model can be derived from a maximum entropy principle under two types of constraints on the statistics of training data ensembles. One is to simulate the birth/death patterns (frequency, position, path, etc.) in the training data, and the other is to match the constraint statistics between objects, such as speed, relative position. β is the set of parameters in the energy,

$$\beta = \{\beta_i(), \beta_k(); \forall i \in (1,2,\ldots,K), \forall k \in \mathcal{R}\}.$$

Each $\beta()$ above is a potential function, not a scalar, and is represented by a vector through discretizing the function in a non-parametric way, as it was done in the FRAME model for texture [16]. Therefore, we can rewrite function (2) as:

$$\varepsilon(\boldsymbol{\mathcal{G}}) = \sum_{\alpha\in\mathcal{R}}\langle\beta_\alpha, H_\alpha(c)\rangle \tag{3}$$

where c is a clique of nodes related with relation α, $H(.)$ is a vectorized function represent statistic property of c. The partition function is summed over all trajectory graph. $Z = Z(\beta) = \sum_{\boldsymbol{\mathcal{G}}}\exp\{-\varepsilon(\boldsymbol{\mathcal{G}})\}$

3. Learning the Prior Model

Given annotated training set sampled from an underlying distribution f governing the motion patterns of object in the observed video sequence $I^{\text{obs}}[0,\tau]$ (suppose the τ is large enough for the motion pattern form an ensemble):

$$D^{\text{obs}} = \{(\boldsymbol{\mathcal{G}}_i^{\text{obs}}) : i = 1,2,\ldots,N\} \sim f(\boldsymbol{\mathcal{G}})$$

$\boldsymbol{\mathcal{G}}^{\text{obs}}$ are from interactive labeling method mentioned in Section 1. The objective is to learn a model p which approaches f by minimizing a Kullback-Leibler divergence.

$$\begin{aligned} p^* &= \arg\min KL(f\|p) \\ &= \arg\min \sum_{\boldsymbol{\mathcal{G}}\in\Omega_{\boldsymbol{\mathcal{G}}}} f(\boldsymbol{\mathcal{G}})\log\frac{f(\boldsymbol{\mathcal{G}})}{p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})}. \end{aligned} \tag{4}$$

This is equivalent to the ML estimate for the optimal relation $\mathcal{R}$ and parameters β,

$$(\mathcal{R},\beta)^* = \arg\max\sum_{i=1}^{N}\log p(\boldsymbol{\mathcal{G}}_i^{\text{obs}};\beta,\mathcal{R}).$$

Learning the probability model includes two phases and all these phases follow the same principle above.

1. Estimating the parameters β from training data D^{obs} for given $\mathcal{R}$
2. Learning and pursuing the relation set $\mathcal{R}$ of $\boldsymbol{\mathcal{G}}$.

In the following, we discuss the two phases respectively.

3.1. Maximum Likelihood Learning of β

For a given relation set $\mathcal{R}$, the estimation of β follow the MLE learning process. Let $\mathcal{L}(\beta) = \sum_{i=1}^{N}\log p(\boldsymbol{\mathcal{G}}_i^{\text{obs}};\beta,\mathcal{R})$ be the log-likelihood, by setting $\frac{\partial\mathcal{L}(\beta)}{\partial\beta} = 0$, we have the following two learning steps.

1. Learning the $\beta_k()$ at each trajectory $k \in (1,2,\ldots,K)$ accounts for its topological events (birth/death). $\frac{\partial\ell(\beta)}{\partial\beta_k} = 0$ leads to the following statistical constraints,

$$E_{p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})}[\mathbf{h}(A(C_k))] = \mathbf{h}_k^{\text{obs}}, \forall k \in (1,2,\ldots,K). \tag{5}$$

In the above equation, $A(C_k)$ are the vectorized function represents for the death/birth events parameters,and $\mathbf{h}(A(C_k))$ is a statistical measure of the parameters, such as the histogram. $\mathbf{h}^{\text{obs}}$ is the observed histogram pooled over all trajectories in $D_{\boldsymbol{\mathcal{G}}}^{\text{obs}}$.

2. Learning the potential function $\beta_\alpha()$ for each relation $r^k \in \mathcal{R}$. $\frac{\partial\ell(\beta)}{\partial\beta_a} = 0$ leads to the following implicit function.

$$E_{p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})}|\,\mathbf{h}(\psi^k(g_k))| = \mathbf{h}_{g_k}^{\text{obs}}, \forall r^k \in \mathcal{R} \tag{6}$$

In the above equation, $\psi^k(g_k)$are the attributes of g_k and $\mathbf{h}(\psi^k(g_k))$ is a statistical measure of the attributes, such as the histogram. $\mathbf{h}_{g_k}^{\text{obs}}$ is the observed histogram pooled over all the subset of g_k in $D_{\boldsymbol{\mathcal{G}}}^{\text{obs}}$.

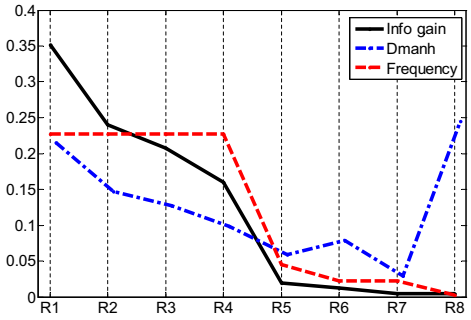

Figure 4. Information measure during the pursuit iterations. Dashed line is the observed frequency of each relation. dotted line is the Mahalanobis distance between $\mathbf{h}^{\text{syn}}$ and $\mathbf{h}^{\text{obs}}$. Solid line is the information gain (KL divergence between p_+ and p). R1=r^{dist}, R2=r^{speed}, R3=r^{smth}, R4=r^{acc}, R5=r^{veh}_{veh}, R6=r^{veh}_{ped}, R7=r^{ped}_{ped}, R8=r^{event}. Relations with both large Mahalanobis distance and high observed frequency are selected first.

The equations (5) and (6) are the constraints for deriving the Gibbs model $p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})$ in equation (1) through the maximum entropy principle.

Due to coupling of the energy terms, equations (6) are solved iteratively through a gradient method. In a general case, we follow the stochastic gradient method adopted in [16], which approximates the expectations $E_{p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})}$ in equation (6) by sample means from a set of synthesized examples.

3.2. Learning and Pursuing the Relation Set $\mathcal{R}$

Besides the learning of parameters β, we can also augment the relation sets $\mathcal{R}$ in a trajectory graph, and thus purse the energy terms in $\sum_{(i,j)\in E_s}\beta_{ij}(\nu_i,\nu_j)$ in the same way as pursuing the filters and statistics in the texture modeling by the minimax entropy principle [16].

Suppose we start with an empty relation set $\mathcal{R}=\emptyset$ thus $p=p(\boldsymbol{\mathcal{G}};\beta,\emptyset)$. The learning procedure is a greedy pursuit. In each step, we add a relation r_+ to $\mathcal{R}$ and thus augment model $p(\boldsymbol{\mathcal{G}};\beta,\mathcal{R})$ to $p_+(\boldsymbol{\mathcal{G}};\beta,\mathcal{R}_+)$, where $\mathcal{R}_+=\mathcal{R}\cup\{r_+\}$.

r_+ is selected from a large pool $\Delta_{\mathcal{R}}$ so as to maximally reduce KL-divergence,

$$\begin{aligned} r_+ &= \arg\max KL(f\|p)-KL(f\|p_+) \\ &= \arg\max \text{KL}(p_+p). \end{aligned} \tag{7}$$

Thus we denote the information gain of r_+ by

$$\begin{aligned} \delta(r_+) &\triangleq KL(p_+\|p) \\ &\approx f^{\text{obs}}(r_+)d_{\text{manh}}(\mathbf{h}^{\text{obs}}(r_+),\mathbf{h}^{\text{syn}}_p(r_+)). \end{aligned} \tag{8}$$

In the above formula, $f^{\text{obs}}(r_+)$ is the frequency that relation r_+ is observed in the training data, $\mathbf{h}^{\text{obs}}(r_+)$ is the histogram for relation r_+ over training data D^{obs}, and $\mathbf{h}^{\text{syn}}_p(r_+)$ is the histogram for relation r_+ over the synthesized trajectory graphs according to the current model p. $d_{\text{manh}}()$ is the Mahalanobis distance between the two histograms.

Intuitively, $\delta(r_+)$ is large if r_+ occurs frequently and tells a large difference between the histograms of the observed and the synthesized trajectory graphs. Large information gain means a significant relation r_+. We refer reader to [16] for more detailed description of the algorithm. Figure 4 shows how the information gain, observe frequency and Mahalanobis distance changes along the pursuing process to the scene described in Figure2.

4. Extended Particle Filter Tracking

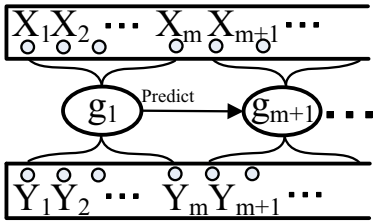

Figure 5. Graphical model for extended particle filter tracking, similar to m-order Hidden Markov Model

Following the general probabilistic framework [1, 8], we formulate a color-based tracking problem by a graphical model illustrated in Figure 5. We denote the target states and image observations in each frame by X_t and Y_t, where $X_t=(d^1_t,d^2_t,...,d^n_t)$, $d^i_t=(x,y)$ are the locations of all the objects at time t. g denotes the state space within m frames: $g_t=(X_t,X_{t+1},...,X_{t+m-1})$. Here we assume g is a stationary random process $p(g_t|\boldsymbol{\mathcal{G}}_{[0:t-m]})=p(g_t|g_{t-m})$, which is true when m is large (100 frames in our experiments). In [8], the 'scale' of tracking bounding box is a variable to be estimated. In our method, however, objects' real world position can be recovered from 2D image using a geometrical calibration method [6] and an object's scale is simply associated with its location. The tracking problem can be formulated as an inference problem with the prediction prior $p(g_t|g_{t-m})$ given by the learned model. We have

$$p(g_t|Y_{[0:(t+m-1)]})\propto p(Y_{[t:(t+m-1)]}|g_t)p(g_t|Y_{[0:(t-1)]})$$

$$p(g_t|Y_{[0:(t-1)]})=\int p(g_t|g_{t-m})p(g_{t-m}|Y_{[0:(t-1)]})dg_{t-m}$$

where $p(Y_{[t:(t+m-1)]}|g_t)$ represent the measurement of observation likelihood (as in [8]), which is a Bhattacharyya distance between the HSV color histograms of the observed object and the reference model.

The state space of g is extremely high ($128^{n\times m}$ if (x,y) is discretized to a $[128\times 128]$ grid) comparing with common particle filtering algorithms (e.g. approximately $128^5\times n$

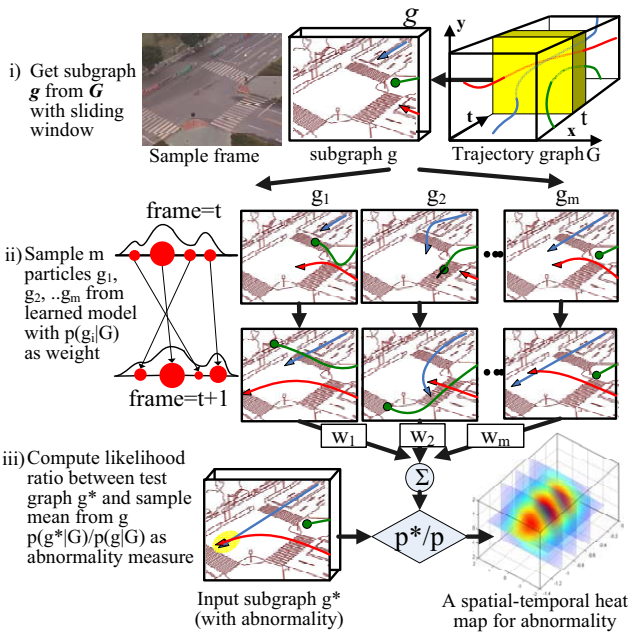

Figure 6. Diagram of sequential importance sampling and abnormal event detection

in [8]). But our method is still computationally achievable, due to the reason that our prediction model $p(g_t|g_{t-m})$ is very close to the true distribution of $p(g_t|Y)$. Thus it is much more effective than a general Markovian dynamics. In practice, we achieve good tracking results (see figure 8) by maintaining only 50 samples (particles).

5. Abnormal Event Detection

We propose a measure to detect any abnormal sub-graph g and identify the part of g that is wrong or violating the traffic regulations, i.e. show that part of g by color (green segments are good, red segments are bad). We have two observations:

(i) whether g is abnormal or not, we need to view g in the context/boundary condition ∂g which is a neighborhood of around g;

(ii) the abnormality is a hypothesis testing problem, we need a "null" model for comparison. Simply computing the conditional probability $p(g|\partial g;\beta)$ won't be right. As we know, when you have a big g, the probability is small, then it is hard to decide on a threshold for different g.

We choose the null model as $p(\mathcal{G};\beta_o)$, for example, $\beta_o = 0$ for the uniform distribution. Therefore we consider the ratio of two conditional probabilities.

$$r(g|\partial g) = \frac{p(g|\partial g;\beta)}{p(g|\partial g;\beta_o)} = \frac{Z(\beta_o)}{Z(\beta)} \exp\{\langle \beta_o - \beta, h(g|\partial g)\rangle\} \tag{9}$$

(Note that we use the vectorized representation given by equation (3) to facilitate computation). This ratio will give us the correct normalization for "how abnormal" the behavior of g is.

The key issue here is to estimate the ratio $\frac{Z(\beta_o)}{Z(\beta)}$. We estimate it by importance sampling. To do so, we need to choose a reference model $p(G;\beta_{\text{ref}})$ and simulate a number of samples from it $g_i \sim p(g|\partial g;\beta_{\text{ref}}), \quad i = 1, 2, ..., N$.

Then we have

$$\begin{aligned}\frac{Z(\beta_o)}{Z(\beta)} &= \frac{\sum_g \exp\{-\langle \beta_o, h(g|\partial g)\rangle\}}{\sum_g \exp\{-\langle \beta, h(g|\partial g)\rangle\}} \\ &= \frac{\sum_g p(g|\partial g;\beta_{\text{ref}}) \exp\{\langle \beta_{\text{ref}} - \beta_o, h(g|\partial g)\rangle\}}{\sum_g p(g|\partial g;\beta_{\text{ref}}) \exp\{\langle \beta_{\text{ref}} - \beta, h(g|\partial g)\rangle\}} \\ &\approx \frac{\sum_{i=1}^N \exp\{\langle \beta_o - \beta_{\text{ref}}, h(g_i|\partial g)\rangle\}}{\sum_{i=1}^N \exp\{\langle \beta - \beta_{\text{ref}}, h(g_i|\partial g)\rangle\}}\end{aligned}$$

Note that we replace the expectation with respect to the reference model by the sample mean using the samples from the reference model. As the reference model could be anything, the key to make the approximation above accurate is that the samples from the reference model overlaps with both models β_o and β. In other words, $g_i, i = 1, 2, ..., N$ from model β_{ref} should be also "typical" for β_o and β.

Usually, if g is large, this approximation is hard to be accurate (e.g. in texture case g will be a patch of images in hi-dimension), it should work well for small g which is true in our case. We choose $\beta_{\text{ref}} = \beta$ the real probability (or the true probability from training data). Plug in, we have

$$\frac{1}{r(g|\partial g)} = \frac{p(g|\partial g;\beta_o)}{p(g|\partial g;\beta)} = \frac{\exp\{\langle \beta, h(g|\partial g)\rangle\}}{\frac{1}{N}\sum_{i=1}^N \exp\{\langle \beta, h(g_i|\partial g)\rangle\}}$$

Or we look at the log-ratio

$$\begin{aligned}-\log r(g|\partial g) &= \langle \beta, h(g|\partial g)\rangle \\ &- \log \frac{1}{N}\sum_{i=1}^N \exp\{\langle \beta, h(g_i|\partial g)\rangle\}.\end{aligned}$$

if g is abnormal, then $r(g|\partial g) \approx 0$, or $-\log r(g|\partial g) >> 1$ That is,

$$\langle \beta, h(g|\partial g)\rangle - \theta >> 1,$$
$$\theta = \log \frac{1}{N}\sum_{i=1}^N [\exp\{\langle \beta, h(g_i|\partial g)\rangle\}].$$

Suppose all the samples have similar (or the same energy) according to β, we can simplify it as

$$\theta \approx \langle \beta, h(g_i|\partial g)\rangle$$

So, the energy of g is much larger than the typical energy of g_i (which are sampled from β and are normal).

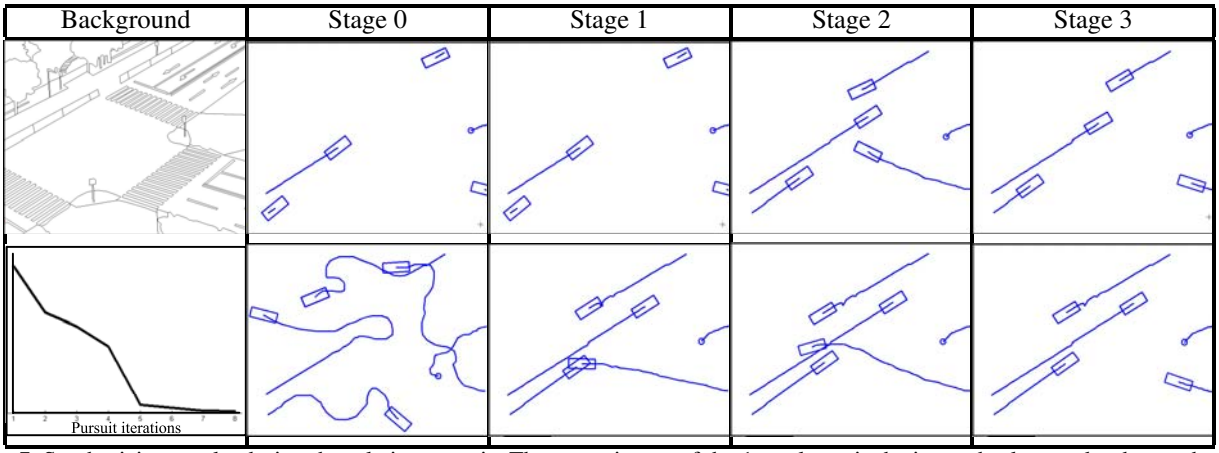

Figure 7. Synthesizing results during the relation pursuit. The upper image of the 1st column is the image background to be synthesized. The lower plot shows the information gain according to learning stages. **Stage 0**: Random motions with no relation constraint. The upper image is the initial frame and the lower image is a synthesized frame (same for the rest columns). **Stage 1**: Synthesized motion with relation [R1-R4] (Definition see figure 4). **Stage 2**: Synthesized motion with relations [R1-R7]. **Stage 3**: Synthesized motion with all relations.

6. Experimental Results

6.1. Experiment I Learning and Synthesizing

We select two traffic surveillance video sequences to test our learning algorithm. Each training video clip is about 30 minutes long and contains about 1,000 objects. The objects are labeled manually. Figure 7 shows synthesized samples from one scene at several stages during the relationship pursuit process. At first the vehicles and pedestrians move randomly around the image following a uniform distribution. At the second stage, relations between moving objects and sources, sinks and paths of the scene are added, which force the objects to move along the tracks, but cannot prevent them from going too fast or colliding into each other. After the speed and pair-wise relations are added, the objects no longer move erratically or hit each other but we can observe cases when vehicles and pedestrians pass crosswalks at the same time. After all the relations are pursued, one can see that the vehicles and pedestrians are no longer moving together.

6.2. Experiment II Object Tracking

We test the proposed tracking method on a challenging surveillance scene and compare the results with the color-based particle filter tracking algorithm described in [8]. The initialization of the particle filter algorithm is given by the ground-truth. Figure8 shows the comparative tracking results for one scene. In this video, only cars are initialized and therefore tracked. We observe that the common particle filter algorithm often quickly collapses to one mode and discards all other modes, therefore fails to track two similar cars after being occluded for a long time period. In the mean time, our algorithm keeps all the modes because the prior term does not allow cars to collide, thus our algorithm does not confuse two similar cars. The overall tracking performance is evaluated by a pixel level precision-recall curve shown in the right part of Figure 8.

6.3. Experiment III. Abnormal Event Detection

Since abnormal events are very rare in observed data, we use three kinds of methods to generate abnormalities: i. tracking failing results (e.g. moving off road, abnormal speed, direction. etc.) ii. collision caused by randomly added trajectories iii. manually added abnormal events such as traffic rule violations. In total, we generate a testing $\mathcal{G}$with 5000 trajectories. As the abnormal event detection is performed by sliding a window on $\mathcal{G}$Figure6, we Then we derive the ground-truth by sliding a sub-window (50 frames) on $\mathcal{G}$and manually label whether there are abnormal events contained. Then we perform the abnormal detection as described in Figure 6. Figure9 shows the detection results with abnormal trajectories marked with red lines and a ROC-curve for these detections.

7. Conclusion and Future Work

We have presented a novel framework for learning motion patterns from observed video sequences and show its ability to detect abnormal behaviors involving multiple objects. The results of sampling this model and using this model for object tracking and abnormal event detection show that it is very useful as prior knowledge of the scene for tracking and object event analysis

Figure 8. Tracking result and comparison. First row: Tracking results with common particle filter method; Second row: Tracking results using particle filter with learned scene contextual prior. Only cars are initialized, therefore the bus and van in the video are not tracked.

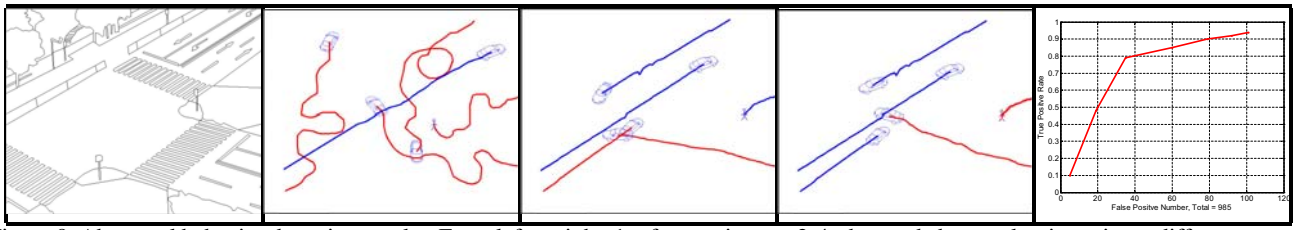

Figure 9. Abnormal behavior detection results. From left to right, 1 reference image. 2-4: detected abnormal trajectories at different stages, 5. ROC-curve of detection results.

8. Acknowledgement

We thank the two reviewers for their suggestions that help improve the presentation of the paper. We are grateful to Dr. Zhuowen Tu for helpful discussions. The work is supported by NSF-DMS 0707055, NSF-IIS 0713652, and ONR N00014-05-01-0543.

References

[1] M. Black and D. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, July 2000. 5

[2] N. P. Cuntoor and R. Chellappa. Epitomic representation of human activities. In *CVPR*, pages 1–8, 2007. 2

[3] A. Hakeem and M. Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, 171:8–9, 2007. 2

[4] F. Jiang, Y. Wu, and A. K. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *ICIP*, pages V: 145–148, 2007. 2

[5] M. P. D. Jolly and A. K. Jain. A modified hausdorff distance for object matching. In *ICPR*, pages A:566–568, 1994. 3

[6] F. Lv, T. Zhao, and R. Nevatia. Camera calibration from video of a walking human. 28(9):1513–1518, 2006. 2, 3, 5

[7] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In *CVPR*, pages II: 955–960, 2005. 2

[8] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. pages I:661–675, 2002. 5, 6, 7

[9] B. Rosario, N. Oliver, and A. Pentland. A synthetic agent system for bayesian modeling of human interactions. pages 342–347, 1999. 2

[10] E. Shechtman and M. Irani. Space-time behavior-based correlation - or - how to tell if two underlying motion fields are similar without computing them? 29(11):2045–2056, Nov. 2007. 2

[11] Y. Shi, Y. Huang, D. Minnen, A. F. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *CVPR*, pages II: 862–869, 2004. 2

[12] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, Aug. 2000. 2

[13] X. Wang, K. Tieu, and W. E. L. Grimson. Learning semantic scene models by trajectory analysis. In *ECCV*, pages III: 110–123, 2006. 1, 2, 3

[14] Y. Z. Wang and S. C. Zhu. Analysis and synthesis of textured motion: Particles and waves. *PAMI*, 26(10):1348–1363, Oct. 2004. 3

[15] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *CVPR*, pages II: 819–826, 2004. 2

[16] S. C. Zhu, Y. Wu, and D. Mumford. Minimax entropy principles and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997. 1, 2, 4, 5