Object Tracking in Structured Environments for Video Surveillance Applications

Junda Zhu, *Student Member, IEEE,* Yuanwei Lao, *Student Member, IEEE,* and Yuan F. Zheng, *Fellow, IEEE*

***Abstract*—We present a novel tracking method for effectively tracking objects in structured environments. The tracking method finds applications in security surveillance, traffic monitoring, etc. In these applications, the movements of objects are constrained by structured environments. Therefore, the relationship between objects and environments can be exploited as additional information for improving the performance of tracking. We use the environment state to model the relationship between the objects and environments, and integrate it into the framework of Bayesian tracking. In this paper, distance transform is used to model the environment state, and particle filtering is employed as the paradigm for solving the Bayesian tracking problem. The adaptive dynamics model and environment prior are devised for the particle filter to fully utilize the environment information in the tracking process. Experiments on some video surveillance sequences demonstrate the effectiveness and robustness of our approach for tracking object motions in structured environments.**

***Index Terms*—Distance transform, object tracking, particle filtering, structured environments, video surveillance.**

I. Introduction

VIDEO SURVEILLANCE has become an indispensable component for ensuring public safety and order in the modern world. Sophisticated video object tracking techniques specially designed for surveillance applications are of increasing importance for analyzing and understanding numerous surveillance videos in an effective manner. A large majority of video surveillance applications are concerned with monitoring activities within structured environments, such as indoor environments, surrounding areas of buildings, highways, traffic junctions, etc., the structures of which are often static and known to the surveillance personnel. One important characteristic of moving objects in these applications is that the motions of objects are constrained by the structure of the environment under surveillance. Therefore, it is beneficial and essential to explore the impacts of the environments upon the object motions, and integrate them into object tracking for improved performances.

Extensive research on generic video object tracking has been done during the last two decades. The body of literature on video object tracking can be categorized into two major classes according to [1]: *object representation and localization*, and *filtering and data association*. The category of *object representation and localization* is represented by the well-known mean shift tracking algorithm which uses simple appearance features, such as color histograms. Based on the kernel density estimate of the appearance features, the mean-shift algorithm iteratively calculates the mean-shift vectors until reaching the local maxima [1]. These algorithms can be considered as local search methods which have the advantage of low computation overheads. Kalman filter and its variants [2], and particle filter [3] are major representations of the filtering and data association method, also known as *Bayesian tracking*. This category of methods solves the object tracking problem by sequentially estimating the state of object using a sequence of noisy measurements about the object states [3].

Object representation and localization methods make little use of the information of the object motions and ignore the environment related factors, and therefore have limited space for improvement by considering the motion characteristics of objects under specific environments. While *filtering and data association* provides a flexible framework for modeling object states and related observations, existing methods do not take the environment factors into consideration, and are therefore not efficient and effective when object motions exhibit strong patterns in structured environments.

There has also been active research on learning based methods for analyzing and understanding surveillance videos [4]–[7]. Stauffer and Grimson [4] proposed a method for learning the activity patterns from surveillance videos. First an object tracker which is based on an adaptive background subtraction method is employed to extract a large set of object motion patterns from surveillance videos over extended periods of time. Then a codebook of activity patterns is generated using an online vector quantization on the whole set of acquired motion patterns. The object tracker of this paper is relatively simple, and although this method gives an interesting solution for learning and classifying object activity patterns under surveillance, it provides little information about motion prediction, which is crucial for improving the efficiency

Manuscript received August 13, 2008; revised April 24, 2009. First version published September 1, 2009; current version published February 5, 2010. This work was supported by the National Science Foundation of USA, under Grant IIS-0328802, and by its Chinese counterpart, the National Natural Science Foundation of China, under Grant 60632040. Research for Y. F. Zheng has been supported by the National Science Foundation, Air Force Research Laboratory, Office of Naval Research, Department of Energy, Dayton Area Graduate Studies Institute, and The Internet2 Technology Evaluation Center of Ohio. This paper was recommended by Associate Editor I. Ahmad.

J. Zhu and Y. Lao are with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: zhuj@ece.osu.edu; laoy@ece.osu.edu).

Y. F. Zheng is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA, and is also with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zheng@ece.osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2031395

and performance of object tracking in video surveillance. Hu *et al.* [5] proposed a method for predicting the object motion and detecting the abnormal activities from surveillance videos, which is based on the learning of statistical motion patterns. An object tracker, which works by clustering foreground pixels using a modified K-means algorithm followed by the growth and prediction of the cluster centroids, is used as the frontend of the system for detecting objects and further extracting object trajectories from the training surveillance videos. Typical motion patterns under given surveillance environments are extracted from training trajectories and represented by chains of Gaussian distributions. Testing trajectories are compared against these motion patterns for the anomaly detection and motion prediction.

Junejo and Foroosh [6], [7] did interesting research on the path learning and modeling for surveillance videos. In [7], camera calibration and trajectory rectification are incorporated into the path modeling system to first remove the projective distortion of the object trajectories. Rectified trajectories are then clustered into distinct paths on which the path model of the surveillance scene is built. The path model is further registered to the aerial imagery to obtain the metric information about object motions, such as the actual moving speed. However, how the numerous trajectories are extracted from surveillance videos is unclear, and how the information of the path model can be used to aid the object tracking in surveillance videos has not been discussed.

Learning based methods are useful for behavior prediction and anomaly detection in video surveillance applications. For learning based methods to yield satisfactory performance, a large number of sample trajectories are required for learning representative motion patterns. Object tracking algorithms act as the frontends of the learning based methods for extracting sample trajectories from a large set of training videos. The accuracy and efficiency of tracking directly affect the performance as well as the computational cost of the trajectory learning methods. Even though existing object tracking methods may be able to track objects for surveillance applications under certain circumstances, ignoring the environment constraints often results in high computational overhead and possibly low tracking accuracy, as can be seen in the experiment part of this paper.

In this paper, we propose a novel approach for tracking video objects in structured surveillance environments for the cases where the statistical knowledge about the object motion patterns is not available. It overcomes the shortcomings of existing object tracking approaches which ignore the environment constraints upon object motions. Considering that the motions of objects are constrained by the environment, we explore the relationship between the objects and the environments as the high-level information to help tracking the objects. We formulate the tracking problem in structured environments in the framework of Bayesian tracking by incorporating the relationship between the objects and the environments as the *environment state* into the state vector of the object. We choose to model the environment state by the distance between the object and environment boundaries, because the state reflects the motion pattern of the object. When objects move approximately parallel to the boundary, such as people walking along the road and vehicles traveling along the lane, such distances are approximately constant, when objects move across regions, such distances are increasing/decreasing gradually. Our approach for solving this particular problem is related to two existing techniques: *distance transform* and *particle filtering*. Environment states are modeled by the distance field using the distance transform and integrated into particle filtering for effective object tracking. Although extensive research has been done on these two individual topics, we are the first to integrate them together for effective object tracking for surveillance videos.

This paper is organized as follows. Section II presents the formulation of our tracking method. Section III studies the details about the environment modeling by the distance field using distance transform. Section IV discusses the integration of the distance field into particle filtering for effective tracking of regular motions as well as covers irregular motions and the mode switching between regular and irregular motions. Section V provides experimental results on the surveillance video sequences, and Section VI concludes the paper.

II. Formulation of the Tracking Problem

In the framework of Bayesian tracking, video object tracking is essentially an estimation problem in which the states of objects are estimated based on noisy observations. The object state is denoted by the state vector $\mathbf{x}_t$, where $t \in \mathbb{N}$ is the time index. The observation is denoted by the observation vector $\mathbf{z}_t$ and represents noisy observations that are related to the state vector [3]. We further denote the sequences of object states and measurements from time t_0 to t_1 by $\mathbf{x}_{t_0:t_1}$ and $\mathbf{z}_{t_0:t_1}$, respectively. From the Bayesian point of view, the problem is to construct the posterior distribution of the state given all available observations, i.e., the joint posterior distribution $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ or the marginal posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$.

A. The Environment State

In the existing works of video object tracking, the state vector only includes the dynamics characteristics of the object, e.g., location, orientation, scale, etc. In this paper, we refer to it as the *dynamics state*, which is denoted by $\mathbf{y}$. For the case of tracking objects in structured environments, object motion is constrained by the environment structure, and the relationship between the object of interest and the environment is also an important characteristic for estimating the object motion. In this paper, we introduce the *environment state*, which models the environment related characteristics of the object, and is denoted by $\mathbf{e}$. The environment state may include, but is not limited to, the geometric relation between the object and environment entities, the property of the region which the object is currently in, etc.

Following the discussion above, the state vector at time t, $\mathbf{x}_t$, for the object within structured environments is comprised of two substates:

1) the dynamics state, $\mathbf{y}_t$;

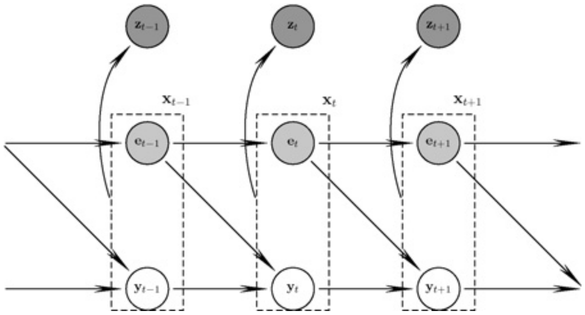


Fig. 1. Graphical probabilistic model for the object tracking with environment states.

2) the environment state, $\mathbf{e}_t$.

With the new formulation of the object state, the probabilistic dependences among variables need to be studied. On the one hand, unlike moving in open spaces the object motion is not only related to its dynamics but also constrained by the environment. In terms of probabilistic dependences, the current dynamics state is dependent upon both previous dynamics states and previous environment states. On the other hand, the current environment state depends upon previous environment states. Current dynamics and environment states are coupled, in our case by the distance map which will be discussed in detail in the following section. Although some works, e.g., [8], have studied the dependences between observations, our paper focuses on improving the tracking algorithm by taking environment constraints into consideration and does not exploit such dependences for simplicity. The probabilistic dependences among the variables are represented by the graphical model shown in Fig. 1. Here we assume the temporal dependences of states to be first order Markovian. These assumptions will be used in the remainder of this paper.

B. Bayesian Updating of the Posterior Probability

Based on both dynamics and environment states, the joint posterior distribution of the object state at time t can be written as $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}) \equiv p(\mathbf{y}_{1:t}, \mathbf{e}_{1:t}|\mathbf{z}_{1:t})$. In order to solve the tracking problem using the recursive Bayesian filters, we first need to express the posterior distribution in a recursively updating form. From the Bayes rule, it follows that

$$\begin{aligned} &p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}) \\ &= \frac{p(\mathbf{z}_t|\mathbf{x}_{1:t}, \mathbf{z}_{1:t-1})p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} && (1) \\ &= cp(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t-1}). && (2) \end{aligned}$$

For (1), we use the conditional independent property $p(\mathbf{z}_t|\mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{x}_t)$, which can be derived from Fig. 1 according to the d-separation property for directed graphs [9], [10]. The denominator of (1) is a constant with respect to the variables $\mathbf{x}_{1:t}$, and is represented by c for clarity.

For the $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t-1})$ term in (2), we have

$$\begin{aligned} &p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t-1}) \\ &\quad \equiv p(\mathbf{e}_{1:t}, \mathbf{y}_{1:t}|\mathbf{z}_{1:t-1}) \\ &\quad = p(\mathbf{e}_t|\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t}, \mathbf{z}_{1:t-1})p(\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t}|\mathbf{z}_{1:t-1}) && (3) \\ &\quad = p(\mathbf{e}_t|\mathbf{e}_{t-1})p(\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t}|\mathbf{z}_{1:t-1}) && (4) \end{aligned}$$

where $p(\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t}|\mathbf{z}_{1:t-1})$ can be further factorized as

$$\begin{aligned} &p(\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t}|\mathbf{z}_{1:t-1}) \\ &\quad = p(\mathbf{y}_t|\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{e}_{1:t-1}, \mathbf{y}_{1:t-1}|\mathbf{z}_{1:t-1}) && (5) \\ &\quad = p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})p(\mathbf{y}_{1:t-1}, \mathbf{e}_{1:t-1}|\mathbf{z}_{1:t-1}) && (6) \end{aligned}$$

by applying the product rule of probability for (3) and (5) and the conditional independent properties for (4) and (6), respectively.

Plug (4) and (6) into (2), and replace the constant terms, the posterior probability of the object state $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ can finally be written as

$$\begin{aligned} &p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}) \\ &\quad = cp(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{e}_t|\mathbf{e}_{t-1})p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})p(\mathbf{x}_{1:t-1}|\mathbf{z}_{1:t-1}) && (7) \end{aligned}$$

which is in a recursively updating form and can be solved by Bayesian filters. The updating terms in (7), i.e., $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})$, $p(\mathbf{e}_t|\mathbf{e}_{t-1})$, and $p(\mathbf{z}_t|\mathbf{x}_t)$, are referred to as the *adaptive dynamics model*, *the environment prior*, and the likelihood, respectively.

III. Environment Modeling Using Distance Transform

Following the formulation of the tracking problem stated above, one of the questions remaining is how the environment state should be modeled. For objects moving in the structured environment, the geometrical relationships between the objects and surroundings are useful observations for estimating object movements. In this section, we introduce the environment modeling approach using *distance transform*. We will first briefly introduce the basic idea of distance transform, and then discuss our approach of environment modeling in detail.

A. Distance Transform

Distance transform (DT) generally refers to the transformation which maps a 2-D binary image into a distance map, in which the value at each point corresponds to its distance to the nearest feature point. It was first developed by Rosenfeld and Pfaltz [11], [12] in the 1960s, and has been widely applied into many image analysis applications, such as shape description [12], skeletonization [13], [14], morphological operations [15], etc. Jarvis [16] successfully extended DT to the problem of robot path planning, in which a collision-free path in a structured environment could always be determined by following the steepest descent direction in the distance map.

Distance transform can be defined as follows in a generic context [17]. Let P be the set of points which we are interested in, and Γ be the set of feature points. The distance transform over the set P associates each point $p \in P$ with the value

$$d(p) = \min_{q \in \Gamma} \mathrm{dist}(p, q) \tag{8}$$

where $\mathrm{dist}(p, q)$ is the distance function which measures the distance between two points p and q. The distance map is a matrix with the same dimension as the transformed image and stores the distance field values $d(p)$ at the corresponding location for each point $p \in P$. A great deal of effort has been made to find efficient algorithms for calculating the distance transform, and there are both exact [18], [19] and approximate [20] algorithms available. Exact algorithms with linear time complexity have been proposed by Breu [18], Maurer [19], etc. Approximate algorithms, often known as chamfer distance transforms, approximate the global distance computation with iterative propagation of the local distance within a small neighborhood mask [20], and therefore have even lower computational costs.

B. Environment Modeling

In order to represent environment constraints using distance transform, we need the geometrical information about the surveillance scene, i.e., how many different regions are there and where the boundaries are located. Static environments are assumed in this paper because the background will not change dramatically over a short period of time for surveillance applications. We first construct the boundary map of the surveillance scene, with the pixels corresponding to region boundaries being ones and otherwise zeros. For surveillance applications, the site plan is often available and can be projected onto the surveillance scene to obtain the boundary map if the scene can be assumed to be planar. The boundary map can also be obtained by image segmentations followed by further adjustments with user inputs when the site plan is not available, such as examples shown in our experiments. We have used the mean shift algorithm [21] for segmentation. The raw outputs of the segmented boundaries are not smooth enough for applying the tracking algorithm we propose in this paper because we need to look up contours and calculate tangential directions of the contour line. Therefore, in our experiments we select seed points from the segmented boundaries and use polynomial curves to fit each region boundary. For structured environments, such an environment model is created once and used for an extended period of time. Thus user-involved segmentation is not a constant overhead. Furthermore, so long as the manual refining is more accurate than the localization of particles, which is true, segmentation will not affect the tracking of the objects because the distance map is to constrain the distribution of the particles, not to introduce additional measurements to the localization of particles. For that reason, how the performance is affected by segmentation is not considered in this paper.

The boundary map $\mathcal{S}$ is divided into R disjoint regions by the boundary pixels Γ. Each region $\Omega_i, i \in 1, 2, \ldots, R$ corresponds to one of the environmental entities, such as roads, lawns, buildings, etc. Applying the Euclidean distance transform [22] upon the binary map, we have

$$d_\Gamma(\mathbf{p}) = \min_{\mathbf{q} \in \Gamma} \|\mathbf{q} - \mathbf{p}\|. \tag{9}$$

The resulting distance field map d_Γ associates each pixel in the scene with a Euclidean distance field value, which corresponds to the shortest distance between this pixel and the boundary, and is related to the distance in the real world by a scale factor depending on the camera model and depth of view.

For the purpose of our tracking application, not only are we interested in the value of the absolute distance, but also want the distance field to distinguish different regions. Because it is not convenient to use region indices in the probabilistic formulation of the environment prior, which will be discussed in detail in the following section, we use weights of different signs to distinguish adjacent regions such that there will be different values on different sides of the boundary. The value of the weighted distance field at a certain point is determined by both the shortest distance to the boundary and the assigned weight of the region to which this point belongs. It can be represented by the following equation:

$$d(\mathbf{p}) = \mathsf{w}(\mathbf{p}) d_\Gamma(\mathbf{p}) \tag{10}$$

where the weight w is assigned according to the properties of the regions, namely

$$\mathsf{w}(\mathbf{p}) = \begin{cases} -1, & \text{if} \quad \mathbf{p} \in \text{pathway} \\ 1, & \text{if} \quad \mathbf{p} \in \text{accessible region excluding pathways} \\ \infty, & \text{if} \quad \mathbf{p} \in \text{inaccessible region.} \end{cases}$$

The above assignment creates a distance field map which associates each point of the scene with a distance field value. Without ambiguity, the weighted distance field is called distance field in the remainder of this paper. For examples in our experiments, pathways refer to roads, alleys and etc., accessible regions refer to regions except pathways, such as lawns, and inaccessible regions are chosen where objects are impossible to appear.

In structured environments, object motion patterns correspond to dynamic processes of environment states. For our case, when an object moves approximately parallel to the region boundary, its distance field values keep near constant; when an object moves across regions, its distance field values are time-varying. Such processes can be estimated by the Bayesian filtering together with dynamics states. We model the environment state by the distance field value d_t together with its velocity $\dot{d}_t = d_t - d_{t-1}$, i.e.,

$$\mathbf{e}_t = [d_t, \dot{d}_t]^T.$$

It should be noted that because the boundary map constructed from image segmentations is based on the unrectified image domain, distance field values are related to metric distances in real world by different scaling factors depending on depths of view. Our modeling of environment states as a time-varying process is able to handle this problem by considering changes of scaling factors in velocity terms of distance field values.

Our approach of environment modeling using the distance transform has the following advantages.

1) The distance map characterizes the topological properties of the environment, and provides an effective way for describing both the geometric relationship between the object and the environment as well as the property of the region which the object is currently in.
2) For the static surveillance scenes, the environment modeling can be done offline and directly applied into the online tracking. The offline computations are simple; the online computation only involves several lookups from the existing distance map, and therefore imposes negligible computational overhead upon the tracking algorithm.

IV. Object Tracking Using Particle Filtering

A. Particle Filtering with Accept–Reject Sampling

Due to the nonlinearity and non-Gaussianity of the state transition and observation models in video object tracking, the posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ cannot be calculated in the closed-form. *Particle Filters*, which are also known as *sequential Monte Carlo methods*, provide approximate solutions to the posterior distributions for which the linearizations and Gaussian assumptions are not applicable. Particle filter has received a great deal of attention in the computer vision area in recent years [23]–[25], since the seminal work of Isard and Blake [23], which first applied particle filter to video object tracking, and is widely known as the CONDENSATION [23] algorithm. In this paper, we use one variant of the particle filter, the *bootstrap filter*, for the task of object tracking in structured environments.

The posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is approximated by a set of N weighted random samples (particles) $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$ with $\sum_{i=1}^N w_t^i = 1$, which can be written as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \sum_{i=1}^{N} w_t^i \cdot \delta(\mathbf{x}_t - \mathbf{x}_t^i). \tag{11}$$

Following our formulation of the tracking problem and the derivation of the bootstrap filter algorithm, the propagation of particles between frames can be drawn from the following conditional distribution:

$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) \propto p(\mathbf{e}_t|\mathbf{e}_{t-1})p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1}) \tag{12}$$

which can also be verified by factorizing the state transition probability $p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ according to the graphical model in Fig. 1. The computation of the particle weights is directly based on the observation likelihood

$$w_t^i \propto p(\mathbf{z}_t|\mathbf{x}_t^i). \tag{13}$$

The minimum mean square error estimation of the object state is the posterior mean, which is given by

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} w_t^i \cdot \mathbf{x}_t^i. \tag{14}$$

As can be seen from (12), the state transition model is related to the environment state, which is consistent with the fact that the object motion in structured environments is under the influence of the environment constraints. However, the state transition model is now in an unnormalized form of the product of two probability distribution functions, and $\mathbf{e}_t$ and $\mathbf{y}_t$ are dependent. Therefore, a rejection sampling scheme is necessary for generating samples according to (12).

Algorithm 1 Accept–Reject Sampling Algorithm

```
repeat
    Sample $\mathbf{x}' \sim g(\mathbf{x})$
    Sample $u \sim \mathcal{U}_{[0,1]}$
until $u \leq \frac{\tilde{f}(\mathbf{x}')}{Mg(\mathbf{x}')}$
Accept the candidate sample $\mathbf{x}'$ as $\mathbf{x}$.
```

As shown in [26], the samples from an unnormalized *target distribution* $\tilde{f}(\mathbf{x})$ can be generated from an *instrumental distribution* $g(\mathbf{x})$ with the only requirement that $\tilde{f}(\mathbf{x}) \leq Mg(\mathbf{x})$ on the support of $\tilde{f}(\mathbf{x})$ where M is a bound on $\frac{\tilde{f}(\cdot)}{g(\cdot)}$, using the accept–reject sampling algorithm below, where $\mathcal{U}$ denotes a uniform distribution.

For our case, the target distribution $\tilde{f}(\cdot)$ corresponds to $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})p(\mathbf{e}_t|\mathbf{e}_{t-1})$, and we choose $g(\cdot)$ to be $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})$. The bound M exists and can be chosen as $\max_{\mathbf{e}_t} p(\mathbf{e}_t|\mathbf{e}_{t-1})$. Rejections of samples are done by the inequality in Algorithm 1, the theoretic proof of which is available in [26]. When a particle with a very small value in $p(\mathbf{e}_t|\mathbf{e}_{t-1})$ is sampled, it has a large probability of rejection for $\frac{\tilde{f}(\mathbf{x}')}{Mg(\mathbf{x}')}$ being smaller than the random number u drawn from the uniform distribution $\mathcal{U}[0, 1]$. Formulations of $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{e}_{t-1})$ and $p(\mathbf{e}_t|\mathbf{e}_{t-1})$ are discussed in detail in the following.

B. The Adaptive Dynamics Model

In this paper, the object at time t is modeled by a rectangular bounding box defined by the dynamics state $\mathbf{y}_t = [m_t, n_t, w_t, h_t, \theta_t, \dot{m}_t, \dot{n}_t, \dot{w}_t, \dot{h}_t, \dot{\theta}_t]^T$, where m, n, w, h, and θ_t correspond to the x/y image coordinates, width, height and orientation, respectively, and components with over-dots are their velocities. The dynamics state transition is subject to the environment constraints, which include the translational constraint and rotational constraint and are discussed as follows.

1) *The Translational Constraint:* Given the estimated object state $\hat{\mathbf{x}}_{t-1}$ at the previous time step, its position $\mathbf{p}_{t-1} = (\hat{m}_{t-1}, \hat{n}_{t-1})$ and distance field value d_{t-1} as well as their velocities are known. Assuming constant velocities for both displacement and distance field, a translational constraint can be devised. On the one hand, the assumed object position at time t should be on the distance field contour $\mathcal{L}_t$ with a value of $d_{t-1}+\dot{d}_{t-1}$, where $\dot{d}_{t-1} = d_{t-1} - d_{t-2}$. On the other hand, the magnitude of displacement at time t should be approximately equal to that of time $t - 1$. The assumed current position of the object, which is denoted by $\mathbf{p}_t^0 = (m_t^0, n_t^0)$, can be written as

$$\mathbf{p}_t^0 = (m_t^0, n_t^0) \in \mathcal{L}_t, \ \text{s.t.} \ \|\mathbf{p}_t^0 - \mathbf{p}_{t-1}\|^2 = \dot{m}_{t-1}^2 + \dot{n}_{t-1}^2 \tag{15}$$

and the translational velocity $\dot{\mathbf{p}}_t^0 = (\dot{m}_t^0, \dot{n}_t^0)$ is updated to be

$$\dot{\mathbf{p}}_t^0 = \mathbf{p}_t^0 - \mathbf{p}_{t-1} \tag{16}$$

which is shown in Fig. 2(a).

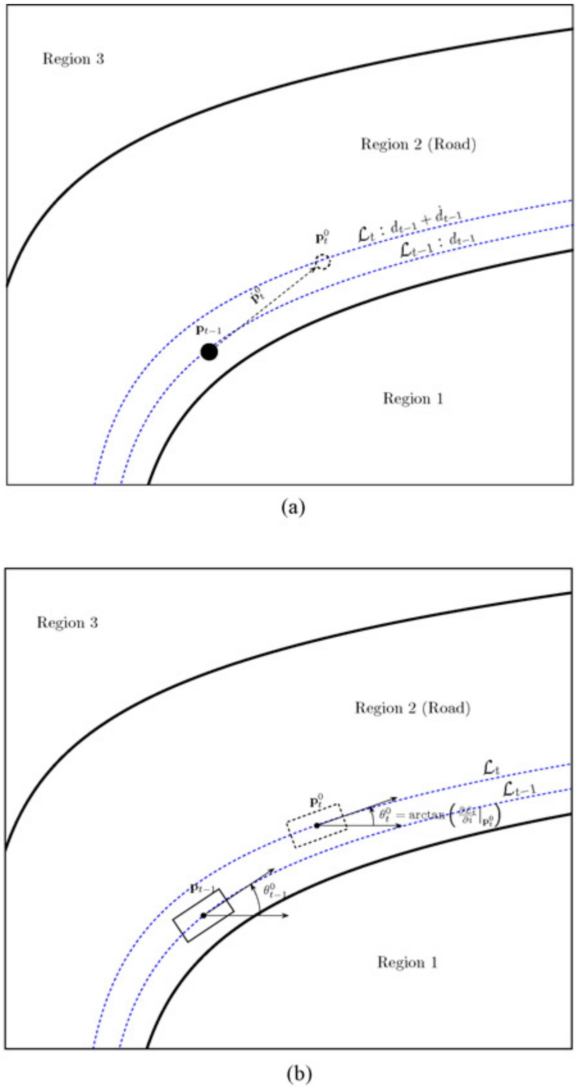


Fig. 2. Constrained motion in structured environments. (a) Translational motion. (b) Rotational motion.

2) *The Rotational Constraint:* Using the translational constraint alone is not enough for nonholonomic objects, such as vehicles, for which orientations of objects are not equal with directions of translations. In this paper, we consider vehicles observed from near top-view angles. Assuming that vehicles follow the direction of road, a rotational constraint can be imposed upon the orientation of the object. Based on the above discussion about the translational motion, the position of the object at time t is likely to be around $\mathbf{p}_t^0$. As a result, the orientation of the object at time t is likely to be along the tangential direction of the distance field contour $\mathcal{L}_t$ at the point $\mathbf{p}_t^0$, as shown in Fig. 2(b). Therefore, the assumed orientation of the object at time t, denoted by θ_t^0, can be written as

$$\theta_t^0 = \arctan\left(\frac{\partial \mathcal{L}_t}{\partial i}\bigg|_{\mathbf{p}_t^0} \right). \tag{17}$$

Let the orientation of the tangent line at $\mathbf{p}_{t-1}$ be θ_{t-1}, the assumed angle velocity of the object can be written as

$$\dot{\theta}_t^0 = \theta_t^0 - \theta_{t-1}. \tag{18}$$

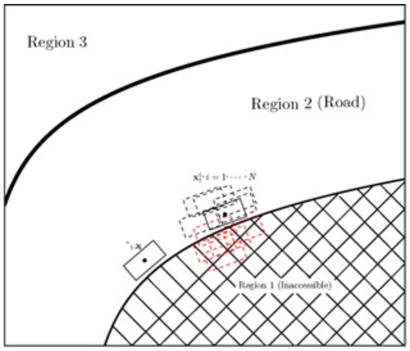


Fig. 3. Example of the distribution of particles near the region boundaries.

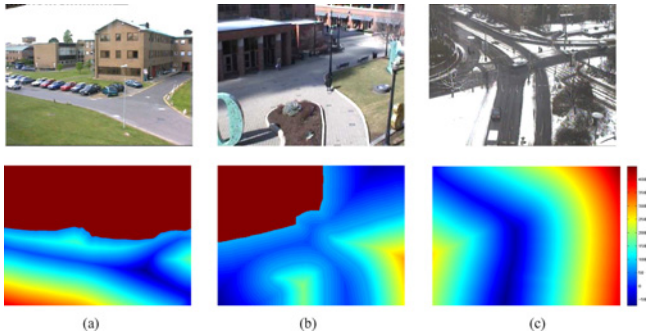


Fig. 4. Examples of surveillance scenes and distance maps used in our experiments. This figure is best viewed in color. (a) PETS2001. (b) OTCBVS. (c) Karl–Wilhelm–Straße.

Following the discussion above, the adaptive dynamics model can be devised. For the ease of the sampling, we choose it to be in the Gaussian form

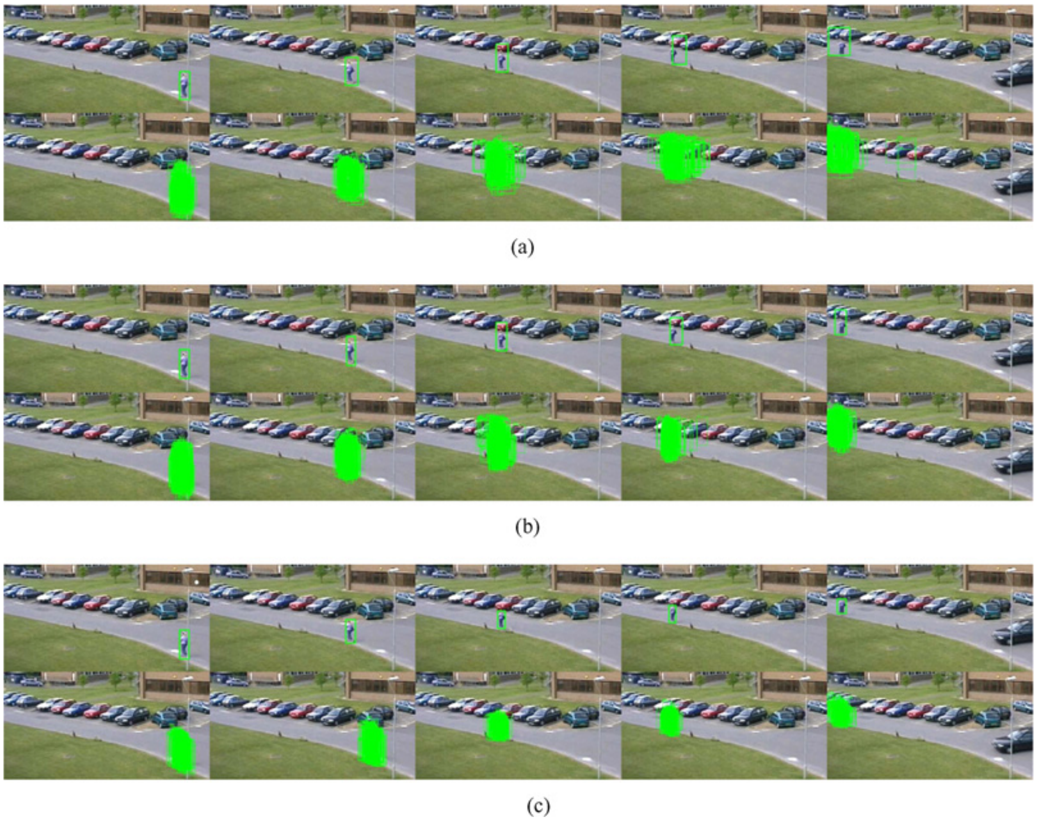
$$\begin{bmatrix} m_t \\ n_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} m_{t-1} \\ n_{t-1} \\ \theta_{t-1} \end{bmatrix} + \mathcal{N}\left(\begin{bmatrix} \dot{m}_t^0 \\ \dot{n}_t^0 \\ \dot{\theta}_t^0 \end{bmatrix}, \begin{bmatrix} \sigma_m^2 & 0 & 0 \\ 0 & \sigma_n^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \right) \tag{19}$$

where $\dot{m}_t^0, \dot{n}_t^0, \dot{\theta}_t^0$ are given by (15)–(18), and update equations of width/height and velocity components are not shown for simplicity.

Note that although we have assumed constant velocities of displacement and distance field, the random sampling can accommodate possible variations of the object motions around the predicted state. Therefore, rather than imposing an overly rigid constraint on the particle propagation, our proposed method provides an informed prediction about the possible state of the object in the presence of environment constraints.

C. The Environment Prior

The adaptive dynamics model described above provides an approximation about the dynamics state in the next frame, around which a large number of particles are randomly distributed. However, without the environment prior, it is possible that some of the particles reside in the regions which the object of interest is unlikely/impossible to visit. An example showing this problem is illustrated in Fig. 3. By using the rejection sampling, few particles would be sampled in the regions with low environment prior values.



(a)

(b)

(c)

Fig. 5. Tracking results of object A on PETS2001 dataset for frames 2149, 2224, 2324, 2399, and 2499. This figure is best viewed in color. (a) GPF with 150 particles. (b) GPF with 300 particles. (c) Our method with 150 particles.

Now the question is how the environment prior density $p(\mathbf{e}_t|\mathbf{e}_{t-1})$ should be modeled. The environment prior should serve for the purpose of regulating particles around possible paths and excluding the impossible regions. Based on the constant velocity assumption, the distance field value d_t at time t should be close to $d_{t-1} + \dot{d}_{t-1}$. We model the environment prior $p(\mathbf{e}_t|\mathbf{e}_{t-1})$ as a normal distribution, namely

$$p(\mathbf{e}_t|\mathbf{e}_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma_e}\exp\left(-\frac{(d_t - d_{t-1} - \dot{d}_{t-1})^2}{2{\sigma_e}^2}\right) \qquad (20)$$

$$= \frac{1}{\sigma_e}\varphi\left(\frac{d_t - d_{t-1} - \dot{d}_{t-1}}{\sigma_e}\right). \qquad (21)$$

Following the formulations of the adaptive dynamics transition and environment prior in (19) and (21), the term $\frac{\tilde{f}(\mathbf{x}')}{Mg(\mathbf{x}')}$ in Algorithm 1 is written as

$$\frac{\tilde{f}(\mathbf{x}')}{Mg(\mathbf{x}')} = \frac{p(\mathbf{e}_t|\mathbf{e}_{t-1})}{\max\limits_{\mathbf{e}_t} p(\mathbf{e}_t|\mathbf{e}_{t-1})} = \varphi\left(\frac{d_t - d_{t-1}}{\sigma_e}\right)\Big/\varphi(0). \qquad (22)$$

Now the particles $\{\mathbf{x}_t^i\}_{i=1,\ldots,N}$ can be readily generated using the accept–reject sampling described in Algorithm 1. A particle candidate with a large value of $|d_t - d_{t-1} - \dot{d}_{t-1}|$, which either has a large deviation from the regular path or resides in an inaccessible region, will be almost surely discarded because $\varphi(\frac{d_t - d_{t-1} - \dot{d}_{t-1}}{\sigma_e})/\varphi(0) \to 0$.

D. Observation Likelihood

In this paper, we use the color histogram model for evaluating the observation likelihood, which is introduced in [27]. For the completeness of the paper, formulations of the observation likelihood model are briefly described as follows.

Given the state $\mathbf{x}_t$ of a particle, the observation likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ is calculated based on the color histogram similarity between the particle and the reference object model

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \exp(-\lambda_b D_b^2(\mathbf{h}(\mathbf{x}_t), \mathbf{h}(\mathbf{x}_0))) \qquad (23)$$

where

$$D_b(h(\mathbf{x}_t), h(\mathbf{x}_0)) = \left(1 - \sum_{b=1}^{b_{max}} \sqrt{\mathbf{h}^b(\mathbf{x}_t) \cdot \mathbf{h}^b(\mathbf{x}_0)}\right)^{\frac{1}{2}} \qquad (24)$$

and $\mathbf{h}(\cdot)$, $\mathbf{h}^b(\cdot)$ represent the color histogram and the value in the bin b of the histogram, respectively. $D_b(h(\mathbf{x}_t), h(\mathbf{x}_0))$ in (24) is known as the Bhattacharyya distance, which measures the similarity between the color histograms of the particle and the reference object model. Details about the calculation of color histogram will not be discussed here, and we refer the readers to [27].

V. Experiment Results

The proposed method has been tested and evaluated on three open surveillance datasets. PETS2001 and OTCBVS datasets are used to test tracking performances on pedestrians, and Karl–Wilhelm–Straße traffic sequence is tested for vehicle tracking. Before presenting detailed results, we first describe the experiment setup, which includes algorithm implementations, model definitions and parameters, tracker initializations, etc.

A. Experiment Setup

Objects are modeled as rectangular bounding boxes. For a pedestrian object, the bounding box has variable width and height, and the orientation is not modeled, i.e., always in the upright orientation. Its distance field value is taken at the bottom center of the bounding box, which is assumed to be on the ground plane. Since the traffic sequence in our test was taken with a near top-view angle, the vehicle object is modeled as a box with a variable orientation. Scale changes of vehicles are not considered to keep the dimensionality of filter state low. Its distance field value is taken at the center of the box.

For the observation model, color histograms are calculated in the RGB space with 16 bins for each channel. The parameter λ_b of the likelihood function is set to be 20 for all the tests.

For the environment modeling, distance maps are obtained by image segmentations followed by manual smoothing. Only regions in which objects will be totally occluded, e.g., buildings, are considered as inaccessible regions. Roads and alleys are marked as pathways, and all other regions, such as sidewalks and lawns are marked as accessible regions. Examples of distance maps used in our experiments are shown in Fig. 4. For the Karl–Wilhelm–Straße sequence, only the part related to our tracking is treated.

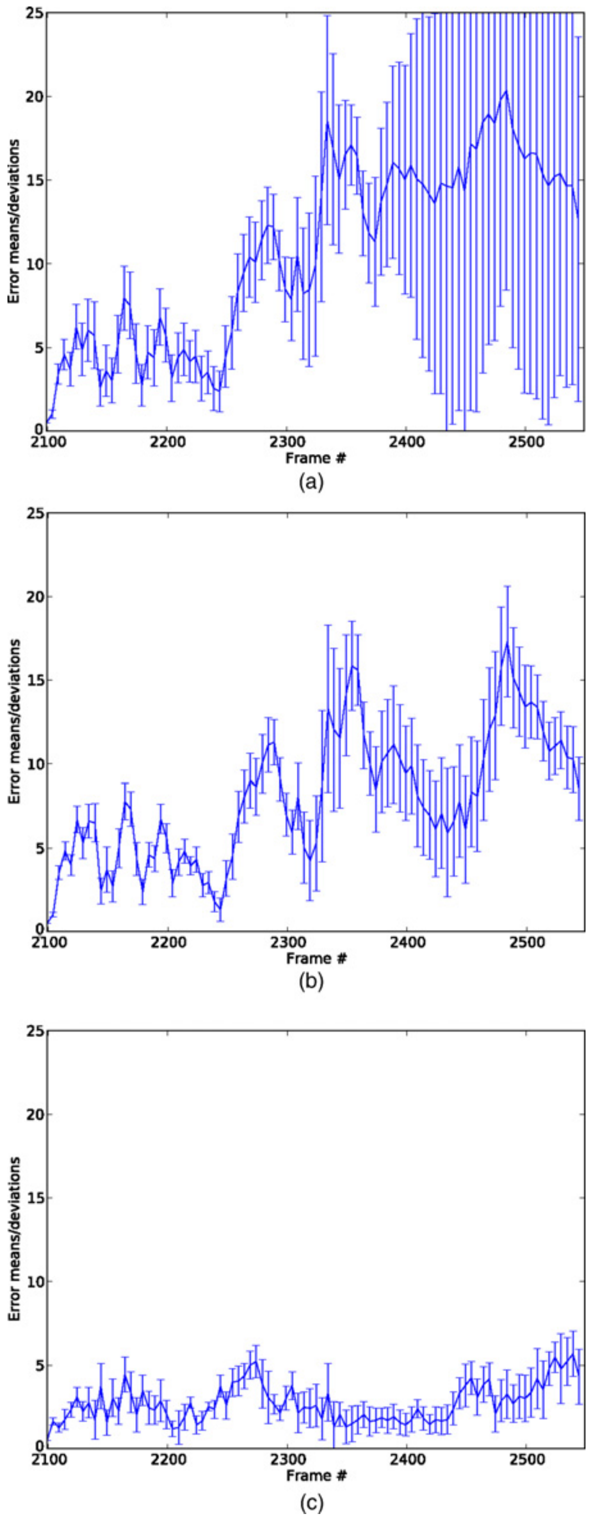
Our algorithm is implemented using the OpenCV library with Python interface. We also implement generic particle filtering trackers with constant velocity models under the same framework as baseline algorithms, which are referred to as GPFs in the rest of the paper. Comparisons are made with respect to GPFs on the tracking accuracy, robustness against clutters, and computational costs in terms of the number of particles.

It should be noted that all trackers are initialized by either utilizing the available ground truth data (for PETS2001), or manually defining the bounding box in the first frame (for the other two datasets). The reference observation template is calculated from the initial state and kept the same throughout the tracking process.

B. Results on PETS2001 Dataset

PETS2001 is a popular surveillance dataset. It provides ground truth data of object trajectories. Dataset 1 includes two surveillance footages from two static cameras shooting the same outdoor environment from different view-angles. For our test, we only use the image sequence from Camera 1, and test our algorithm on several pedestrian objects. Both qualitative and quantitative results are shown for this dataset.

The first test case is to track a single person (referred to as object A) walking along a curved road. The original image sequence is down-sampled to 5 frames/s for testing. Fig. 5 shows tracking results of GPFs with 150 and 300 particles, and our method with 150 particles, respectively. As the tracking goes, the GPF with 150 particles shows increasingly large errors in both object locations and scales. Increasing the number of particles to 300 for GPF improves the tracking performance, however, scale changes of object are still not well estimated. Using only 150 particles, our method shows satisfactory tracking results for both object locations and scales.



(a)

(b)

(c)

Fig. 6. Means and deviations of tracking errors of object A. Results are calculated from 100 repeated runs with a frame rate of 5 frames/s. (a) GPF with 150 particles. (b) GPF with 300 particles. (c) Our method with 150 particles.

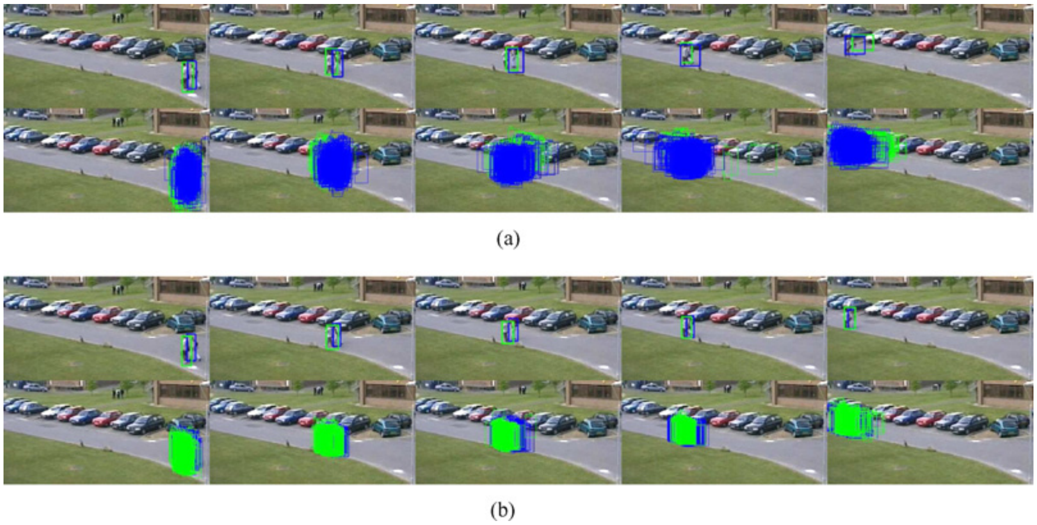


Fig. 7. Tracking results of objects B and C using two independent trackers at a frame rate of 5 frames/s. Object B: shown in green boxes. Object C: shown in blue boxes. Frames shown: 1487, 1587, 1637, 1687, 1787. This figure is best viewed in color. (a) GPFs with 300 particles each. (b) Our method with 150 particles each.

To further evaluate the performance of our proposed method in a quantitative manner, we repeat experiments for 100 times and calculate means and deviations of tracking errors against the ground truth associated with the dataset. The tracking error is defined as the displacement in pixels, between the ground truth centroid of the object and the center of tracked bounding box. Fig. 6 shows error bar plots of GPFs with 150 and 300 particles and our tracker with 150 particles, respectively. Tracking errors of GPFs with 150 and 300 particles have similar trends in their means, and large increases in average errors are observed around frames 2240 to 2290 and 2320 to 2360 when the object is turning along the curved road or close to some background clutters. Error deviations of the GPF with 150 particles diverge near the end of tracking, which indicates that the GPF is unable to provide reliable tracking results as 150 particles are not enough to effectively cover the state space without utilizing environmental constraints. Our proposed method shows smaller error means and deviations than GPFs. The average error also has smaller fluctuations compared to those of GPFs, which can be explained that our adaptive dynamics model fits the motion of the object better than the constant velocity model when the object being tracked adjusts its motion from time to time under environmental constraints.

The second test case is to track two persons walking together. We refer to the person in the front as object B, results of which are shown in green, and the one in back as object C, results of which are shown in blue. It is more challenging than the first case since two objects have similar color tops (white versus cream) and occasional occlusions. We use two independent trackers with no explicit interactions between them. Tracking results of GPFs with 300 particles for each object are shown in Fig. 7(a). The tracking result of object B is unstable. Around frame 1637, objects are moving near some vehicles with similar color distributions to part of object B. Particles are attracted by these background clutters, and tracking errors start to accumulate. Although particles are able to escape from these clutters later, they are not able to recover to the true object, and therefore tracking results afterward are largely inaccurate. Tracking results of object C show large errors in its widths. The bounding box of C coalesces onto both objects after temporal occlusions. This is because the nondistinctive appearances of two objects and no explicit interactions between two trackers. Fig. 7(b) shows the tracking results of our method with 150 particles for each tracker. Tracking results of object B are quite stable, and particles are kept off from background clutters and effectively cover the object path. Tracking results of object C also see considerable improvements over those of the GPF, although are a little affected by occlusions. Although tracking results also show tendencies of covering both objects instead of one near the end of the tracking, as objects become smaller and feature less distinctive, our method manages to keep separate tracking of the two objects most of the time without explicit handling of occlusions and data associations. Fig. 8 shows the curves of distance field values along the object paths throughout tracking, where absolute values are shown instead of negative weighted distances for clarity. Both curves show quick decreasing at the beginning when objects move toward the boundary, and fluctuate slowly after objects move parallel to the boundary. By introducing the distance velocity in the object state, the tracker can adapt itself to the changes of the distance field values. Distance field values of object C are higher than those of object B as it is further from the boundary all the time, which implies relative positions between two objects and helps resolve slight occlusions between objects for this case.

We have also studied the impact of different frame rates on the performance of tracking algorithm. Three different frame rates, 12.5/5.0/2.5 frames/s, are tested on both GPF and our method about tracking accuracies of objects B and C. Experiments are repeated for 25 times for each case, and average tracking errors are shown in Fig. 9. Our method has

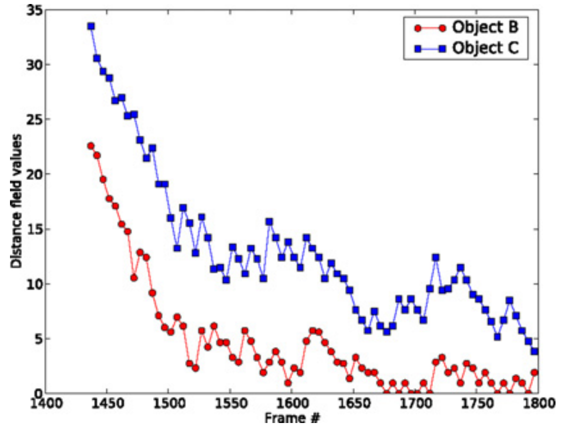


Fig. 8. Distance paths of objects B and C. For clarity, absolute values of weighted distance field are shown.

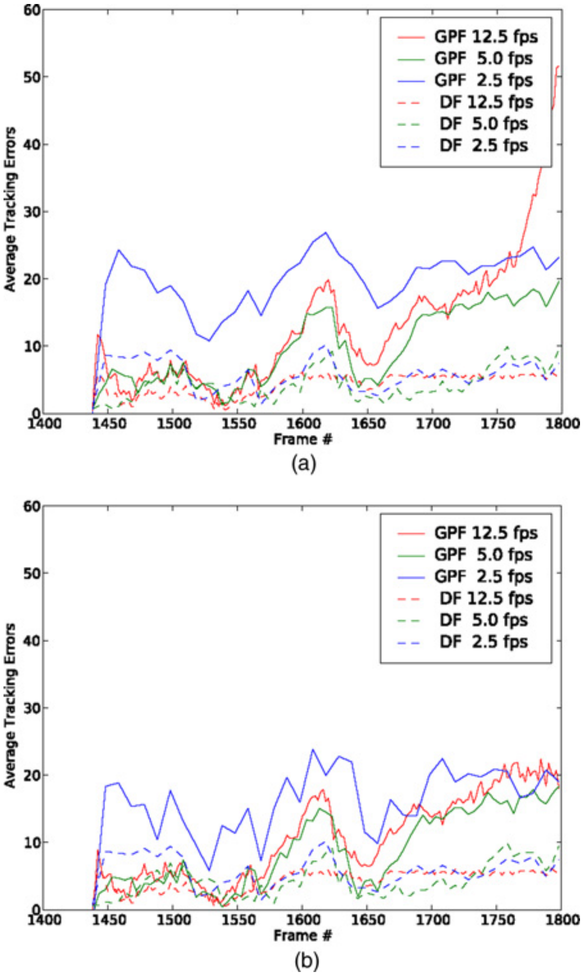


Fig. 9. Average tracking errors of objects B and C for PETS2001 sequence with different frame rates. Averages are calculated from 25 repeated runs. (a) Object B. (b) Object C.

low errors for all three frame rates, and performs similarly for both objects. GPF yields large tracking errors for the low frame rate of 2.5 frames/s for both objects. For 5.0 and 12.5 frames/s cases, GPF shows similar performances, and the error for object B even diverges for 12.5 frames/s tests. This is because for the higher frame rate, the object has smaller displacements between frames. It is more difficult for particles to escape from background clutters once being trapped around them.

C. Results on OTCBVS Dataset

This dataset is taken from IEEE OTCBVS WS series bench [28], which was recorded on OSU campus and can be found at http://www.cse.ohio-state.edu/OTCBVS-BENCH/bench.html. The sequence is recorded by a stationary surveillance camera at a resolution of 320 × 240, and is downsampled to 5 frames/s in our tests. Since there is no ground truth data available, qualitative results are shown for this dataset. Two scenarios are illustrated: the first one recording a person walking along one side of the alley, i.e., keeping almost equal distance to the boundary, and the second one recording a person first walking along one side then crossing to the other side.

Fig. 10 shows the tracking results of the first scenario. Similar as the single object tracking from the PETS2001 dataset, our proposed method uses half of the particle number to achieve similar results with the GPF, as particles are concentrated around high likelihood regions in the state space, and fewer particles are needed as a result.

The second scenario is used to test whether our method can keep continuous track of the object when its distance from the boundary first increases and then decreases. Fig. 11 shows tracking results of our method with 150 particles. In frames 1300 and 1400, tracking results are slightly inaccurate. However, the tracker is able to recover and gives accurate results afterward. It shows that by modeling the environment state as a dynamic process and including it in the particle filtering framework, its changes can also be captured.

D. Results on Karl–Wilhelm–Straße Traffic Dataset

The test video sequences for vehicle tracking are taken from the online image sequence database at http://i21www.ira.uka.de/image_sequences. We have used two color sequences with a resolution of 768 × 576 pixels at 25 frames/s, which monitor the traffic of the same road during heavy fog and snow conditions. These two sequences are referred to as *Fog Traffic* and *Snow Traffic* in the following discussions.

For the *Fog Traffic* sequence, we are to track a car moving along a curved road. Tracking results of GPF and our method are shown in Fig. 12, which are zoomed to the region of interest. Particle numbers of GPF and our method are 300 and 150 respectively. Both trackers can continuously track the car; however, with fewer particles, our method has better accuracy than the GPF. In frames 20 and 40, the car is steering along the curved road segment. Tracking results of GPF in these two frames deviate from true positions and orientations of the object, because the dynamics model cannot adapt itself to environment constraints. With the adaptive dynamics model and environment prior, our tracker can effectively capture the motion of the car, and therefore achieves better accuracies in tracking results.

For the *Snow Traffic* sequence, we are to track a van of white color driving along the road with snow covered surroundings. Considering that we have used the color histogram as the

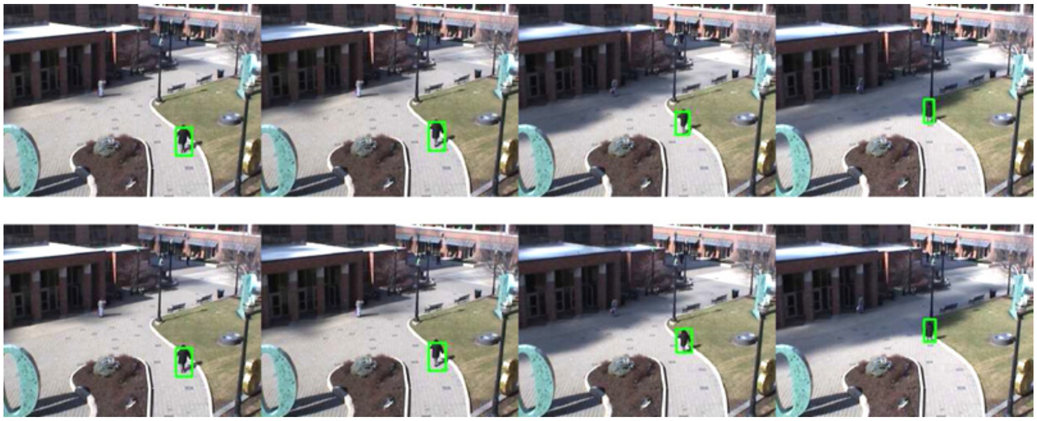


Fig. 10. Tracking results on OTCBVS sequence for object 1. Frames shown: 50, 100, 200, 300. (a) GPF with 300 particles. (b) Our method with 150 particles.

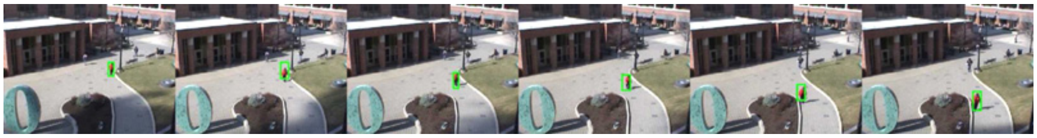


Fig. 11. Tracking results of our method with 150 particles on OTCBVS sequence for object 2. Frames shown: 1200, 1300, 1360, 1400, 1540, 1600.

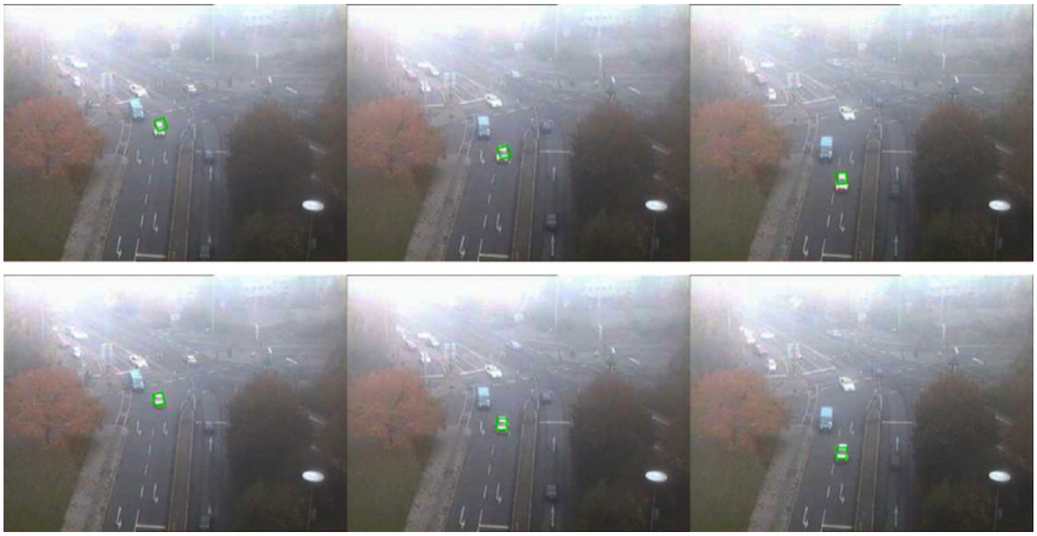


Fig. 12. Tracking results on *Fog Traffic* sequence. (a) GPF with 300 particles. (b) Our method with 150 particles. Frames shown: 20, 40, 60.

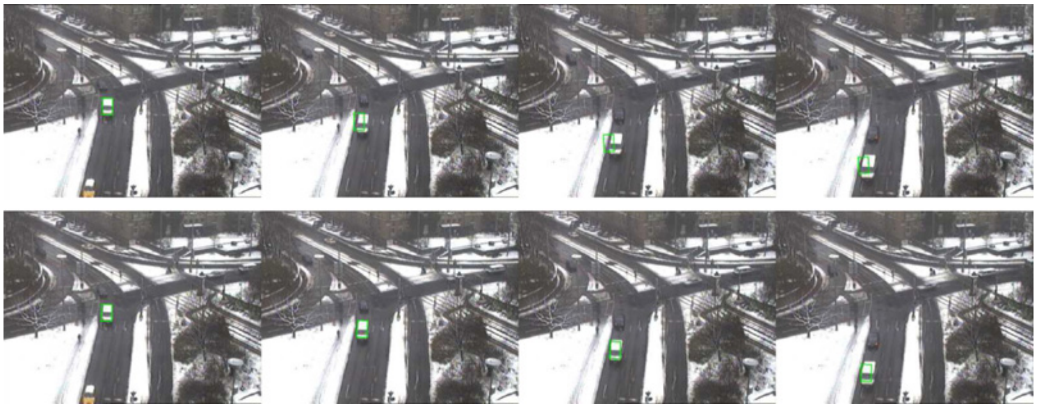


Fig. 13. Tracking results on *Snow Traffic* sequence. (a) GPF with 300 particles. (b) Our method with 150 particles. Frames shown: 220, 240, 260, 280.

observation model, this sequence challenges the robustness of the tracking algorithms against background clutters. In our experiments, numbers of particles for both trackers are kept the same as in the *Fog Traffic* sequence. Fig. 13 illustrates tracking results of both trackers. In frame 220, both trackers are tracking the object as the color of surrounding is quite different from that of the object. However, when the object is approaching snow covered surroundings, performances of GPF and our method are different. Tracking results of GPF have large errors and are at the risk of losing track at the end of the sequence, because a large portion of particles are attracted by surrounding regions which have similar color with the object. This result illustrates the advantage of utilizing the environment constraints with the presence of environment clutters.

VI. Conclusion

In this paper, we have presented a novel tracking method for effectively tracking objects in structured environments for video surveillance applications. Inspired by the fact that the motions of objects in structured environments are constrained by the environment and the relationship between the objects and the environments can be used as extra information for estimating the object motions, we introduced a novel environment state term, which is modeled by the distance field using distance transform, into the Bayesian tracking framework, and solve the tracking problem using the particle filter. By this approach, we were able to integrate the impact of the environment upon object motions into the tracking algorithm. Experimental results demonstrated that the proposed method can greatly improve the performance of tracking in the surveillance video. The proposed tracking method has following features.

1) The dynamics model of the particle filter is adaptive to the local structure of the environment, i.e., the distance map. It provides an informed assumption about the object dynamics based on the environmental context.
2) The rejection sampling based on the environment prior helps generating particles to cover the regions which the object is most likely to visit, and avoid the unlikely and inaccessible regions.

As mentioned in the beginning of this paper, the proposed method intends for structured environments and relies on the availability of environment models. For structured environments, the environment model can be obtained using segmentation methods which have been extensively studied in the computer vision field. In our approach, we employ the mean shift algorithm for segmentation which is followed by a manual refinement of the boundaries to generate smooth contours for creating the distance map. Experimental results prove that using such an environment model, we are able to produce better results than the generic particle filter approach. Since the map can be used for an extended time period for surveillance applications, the manual involvement in the segmentation process is not a constant overhead.

The proposed method can be extended to complex and dynamic changing environment provided an effective and efficient modeling approach for 3-D space is available, which will be the topic for our future studies. Since modeling objects by bounding boxes will not be appropriate for more complex cases, e.g., vehicles not view from near top-view angles, more sophisticated modeling of objects is another topic of study in the future. Occlusion handling and data association for multiple object tracking is the third topic of study for our future research. With all the three topics resolved the constrained object tracking as proposed in this paper can be applied to complex and dynamic environments.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel based object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, 1st ed. New York: Wiley, 2001.

[3] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, 1st ed. Norwood, MA: Artech House, 2004.

[4] C. Stauffer and W. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[5] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, “A system for learning statistical motion patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[6] I. N. Junejo and H. Foroosh, “Euclidean path modeling from ground and aerial views,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–6.

[7] I. N. Junejo and H. Foroosh, “Trajectory rectification and path modeling for video surveillance,” in *Proc. IEEE 11th Conf. Comput. Vision*, Oct. 2007, pp. 1–7.

[8] J. Odobez, D. Gatica-Perez, and S. O. Ba, “Embedding motion in model based stochastic tracking,” *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3514–3530, Nov. 2006.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. New York: Springer-Verlag, 2006.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1st ed. San Francisco, CA: Morgan Kaufmann, 1988.

[11] A. Rosenfeld and J. Pfaltz, “Sequential operations in digital picture processing,” *J. ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1966.

[12] A. Rosenfeld and J. Pfaltz, “Distance functions on digital pictures,” *Pattern Recognit.*, vol. 1, no. 1, pp. 33–61, Jul. 1968.

[13] Y. Chehadeh, D. Coquin, and P. Bolon, “A skeletonization algorithm using chamfer distance transformation adapted to rectangular grids,” in *Proc. 13th Int. Conf. Pattern Recognit.*, vol. 2. Aug. 1996, pp. 131–135.

[14] L. J. Latecki, Q. N. Li, X. Bai, and W. Y. Liu, “Skeletonization using SSM of the distance transform,” in *Proc. IEEE Conf. Image Process.*, vol. 5. Sep. 2007, pp. 349–352.

[15] P. Maragos, “Differential morphology and image processing,” *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 922–937, Jun. 1996.

[16] R. Jarvis, “Distance transform based path planning for robot navigation,” in *Recent Trends in Mobile Robots*, Y. Zheng, Ed., 1st ed. Englewood Cliffs, NJ: World Scientific, 1993.

[17] S. Marchand-Maillet and Y. M. Sharaiha, *Binary Digital Image Processing: A Discrete Approach*, 1st ed. New York: Academic, 2000.

[18] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, “Linear time Euclidean distance transform algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 5, pp. 529–533, May 1995.

[19] C. R. J. Maurer, R. Qi, and V. Raghavan, “A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 265–270, Feb. 2003.

[20] M. A. Butt and P. Maragos, “Optimum design of chamfer distance transforms,” *IEEE Trans. Image Process.*, vol. 7, no. 10, pp. 1477–1484, Oct. 1998.

[21] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[22] G. Borgefors, “Distance transformations in digital images,” *Comput. Vision Graph. Image Process.*, vol. 34, no. 3, pp. 344–371, Jun. 1986.

[23] M. Isard and A. Blake, "Condensation: Conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[24] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1491–1506, Nov. 2004.

[25] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of multiple objects in color video sequences," in *Proc. 8th Int. Conf. Information Fusion*, vol. 1. Jul. 2005, p. 7.

[26] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer-Verlag, 2005.

[27] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color based probabilistic tracking," in *Proc. 7th Eur. Conf. Comput. Vision*, LNCS 2350. Jun. 2002, pp. 661–675.

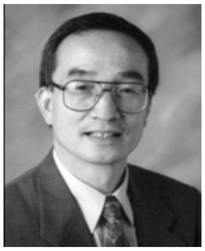
[28] J. Davis and V. Sharma, "Fusion based background-subtraction using contour saliency," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 3. Jun. 2005, p. 11.

Junda Zhu (S'08) received the B.S. and M.S. degrees in information science and electronics engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Ohio State University, Columbus.

His research interests include the field of image and video processing, with an emphasis on video object tracking.

Yuanwei Lao (S'06) received the B.S. and M.S. degrees in information science and electronics engineering from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Ohio State University, Columbus.

His research interests include image and video processing, pattern recognition, and their multimedia applications.



Yuan F. Zheng (F'97) received the B.S. degree in engineering physics from Tsinghua University, Beijing, China, in 1970, and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University (OSU), Columbus, in 1980 and 1984, respectively.

From 1984 to 1989, he was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC. He was the Chairman of the Department of Electrical and Computer Engineering, OSU, from 1993 to 2004, where he has been since 1989. Currently, he is a Professor at the Department of Electrical and Computer Engineering, OSU. He is also with Shanghai Jiao Tong University, Shanghai, China. From 2004 to 2005, he spent a sabbatical year at the Shanghai Jiao Tong University, Shanghai, China, where he continued to be involved as the Dean of the School of Electronic, Information and Electrical Engineering for part-time administrative and research activities until 2008. He is on the editorial board of five international journals. His research interests include image and video processing for compression, object classification, object tracking, and robotics for which his current activities are in robotics and automation for high-throughput applications in biology studies.

Dr. Zheng received the Presidential Young Investigator Award from Ronald Reagan in 1986, and the Research Award from the College of Engineering of OSU in 1993, 1997, and 2007. He and his students received the Best Student Paper and Best Conference Paper Awards several times, and received the Fred Diamond Award for the Best Technical Paper from the Air Force Research Laboratory in 2006.