Integrating Spatio-Temporal Context with Multiview Representation for Object Recognition in Visual Surveillance

Xiaobai Liu, Liang Lin, Shuicheng Yan, *Senior Member, IEEE,* Hai Jin, *Senior Member, IEEE,* and Wenbing Tao

***Abstract*—We present in this paper an integrated solution to rapidly recognizing dynamic objects in surveillance videos by exploring various contextual information. This solution consists of three components. The first one is a *multi-view object representation*. It contains a set of deformable object templates, each of which comprises an ensemble of active features for an object category in a specific view/pose. The template can be efficiently learned via a small set of roughly aligned positive samples without negative samples. The second component is a *unified spatio-temporal context model*, which integrates two types of contextual information in a Bayesian way. One is the spatial context, including main surface property (constraints on object type and density) and camera geometric parameters (constraints on object size at a specific location). The other is the temporal context, containing the pixel-level and instance-level consistency models, used to generate the foreground probability map and local object trajectory prediction. We also combine the above spatial and temporal contextual information to estimate the object pose in scene and use it as a strong prior for inference. The third component is a robust *sampling-based inference procedure*. Taking the spatio-temporal contextual knowledge as the prior model and deformable template matching as the likelihood model, we formulate the problem of object category recognition as a maximum-a-posteriori problem. The probabilistic inference can be achieved by a simple Markov chain Mento Carlo sampler, owing to the informative spatio-temporal context model which is able to greatly reduce the computation complexity and the category ambiguities. The system performance and benefit gain from the spatio-temporal contextual information are quantitatively evaluated on several challenging datasets and the comparison results clearly demonstrate that our proposed algorithm outperforms other state-of-the-art algorithms.**

Manuscript received October 4, 2008; revised May 5, 2009 and August 15, 2009; accepted June 3, 2010. Date of publication October 14, 2010; date of current version April 1, 2011. This work was supported by the CSIDM Project CSIDM-200803, partially funded by a grant from the National Research Foundation administered by the Media Development Authority of Singapore. This work was also supported in part by the National High Technology Research and Development Program of China, under Grant 2006AA01A115, and in part by the National Natural Science Foundation of China, under Grant 60970156. This paper was recommended by Associate Editor I. Ahmad. (Corresponding author is Liang Lin).

X. Liu, H. Jin, and W. Tao are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: elelxb@nus.edu.sg; hjin@hust.edu.cn; wenbingtao@hust.edu.cn).

L. Lin is with the School of Software, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: linliang@ieee.org).

S. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: eleyans@nus.edu.sg).

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.2010.2087570

***Index Terms*—Active feature, deformable template, object recognition, spatio-temporal context.**

I. Introduction

VISUAL SURVEILLANCE is a hot research topic in computer vision owing to its great industrial application potentials. One of the core tasks for a surveillance system is to rapidly recognize objects in video sequences with the presence of various challenges, such as sudden light changing, occluding, and so on. Many related efforts [7], [13], [25] have been proposed from two major aspects: 1) how to develop efficient and effective appearance features, and 2) how to integrate scene contextual knowledge as prior cues.

In this paper, we address the above two problems and particularly highlight the contextual information for visual surveillance applications. The advantages of harnessing spatial and temporal context are illustrated in Fig. 1. Given an observed scene, the patches of interest (highlighted as the patches "a," "b," and "c") can be recognized based on the valuable cues from spatial context (camera geometry, scene surface knowledge, or other surrounding objects). For the case with object occlusions (the tth frame and last frame), the recognition task for the patches "d" and "e" can only be achieved based on the temporal context from the deferred observations, e.g., the temporal consistence of appearance over the consecutive frames from $t-\tau$ to $t+\tau$. Taking full advantage of various contextual information, we present in this paper a flexible solution to real-time recognizing and localizing moving objects in videos, by integrating spatio-temporal contextual information with a novel object detector via deformable template matching.

A. Related Works

There has been a wide variety of works for rapid object category recognition in videos. According to the information sources used for this task, we roughly divide these previous efforts into two categories, namely, the appearance information and the contextual information.

Appearance information is captured by various appearance features, including texture-based features, such as scale invariant points/patches [9], [21], texton filter responses [15], and colorized distributions [3], and structure(shape)-based features, such as shape filter responses [22], edge/ridge fragments [8], active contour, and deformable templates [34]. Based on those features, quite a number of category recognition approaches

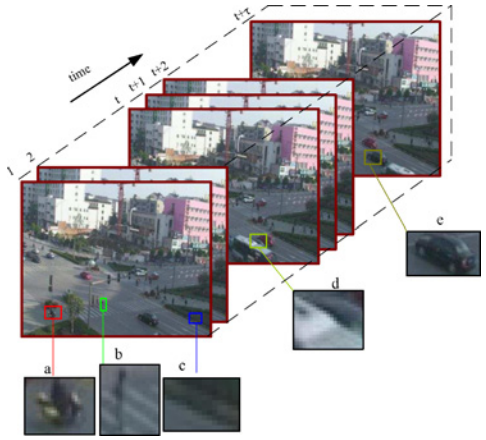

Fig. 1. Illustration of the benefit from contextual information. It is generally impossible to recognize the (a) bike, (b) pedestrian, or (c) lawn separately. We can, however, take them together to form a coherent visual story and combine with the camera and scene surface knowledge to achieve these missions. Furthermore, for the case with occlusions, the recognition of (d) patch is infeasible without the additional temporal context from the (e) deferred observations. Therefore, both spatial and temporal contexts are very useful for object category recognition in low-resolution surveillance videos.

have been well studied, such as the bag-of-words model [30], K-fans model [8], constellation model [24], [26], and boosting family [2], [22], [33], [36]. However, they often fail in surveillance applications, because: 1) the scenes/objects in the surveillance systems are usually in low-resolution, and thus neither local appearance nor part-based configuration are informative enough [31], and 2) moving objects usually undergo with pose/view changes, and thus the classifiers (such as the deformable template matching) need to be re-activated from the classifier ensemble (dictionary). Addressing these two difficulties, in this paper, we proposes to use a multiview representation via deformable object template for category recognition task in realtime videos.

Contextual information has proved to be a critical component in object recognition task, with strong psychophysical evidence. In the computer vision literature, several classic approaches [18] model the relation between the objects of interest and the scene configuration in a graphic representation, (e.g., the knowledge on the location of a road may influence the detection of vehicles [18]), learn implicit inter-related features for objects co-occurrences (conditional/discriminative random field family) [15], [29], or extract scene constraints for false alarm pruning [10], [12]. These approaches have been applied for objects detection/segmentation in cluttered images and achieved impressive performances compared to those ignoring contexts. To our best knowledge, the contextual information has however not been extensively studied for visual surveillance systems. Our current work aims to elicit various contextual information for realtime recognition task, motivated from the following observations.

1) The spatial context, e.g., camera geometry and major surfaces properties etc., contains multiple semantic knowledge and thus can provide prior constraint for recognizing objects in surveillance videos. Also, the context information can be modeled efficiently in interactive ways.

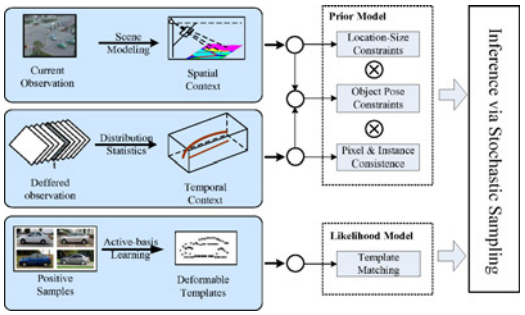

Fig. 2. Comprehensive illustration of the proposed solution for rapidly recognizing dynamic objects in surveillance videos. Within the Bayesian framework, the stochastic inference combines four terms: location-size constraint (from spatial context), pixel and instance-level consistence constraint (from temporal context), object pose constraint (from both spatial and temporal context), and matching verification (based on deformable templates).

2) The temporal context should be also accounted. For example, the recognition in the current frame can be influenced by the previous and the later observations.
3) The spatial and temporal contexts can be further combined to produce more informative constraints on moving objects, e.g., the possible object pose at the specific location of the scene. It is worth noting that our contextual model does not integrate the inter-object co-occurrence relation, which is widely used in previous works [1], [16], since the co-occurrence frequency usually contains little discriminative information for extracting foreground objects in surveillance videos.

This paper focuses on a unified spatio-temporal contextual modeling and view-independent object category representation. Two closest works to ours are the active basis theory proposed by Wu *et al.* [34] and the perspective geometric context model by Hoiem *et al.* [10]. The former is a generative learning theory for image pattern modeling, and the proposed detector in this paper is an extension. The latter is a framework for placing local static object detection within the context of the overall 3-D scene, and we extend it in a unified way for visual surveillance system to improve context modeling.

B. Overview of Our Method

Given a frame from the observed video sequence, we classify and localize moving objects by integrating the spatio-temporal context model and deformable template matching. In general, the context components can be viewed as informative constraints or proposals, and the deformable template matching component can be viewed as top-down verification. We describe the entire solution from three aspects as below.

1) *Multi-View Object Representation via Deformable Templates:* We learn a deformable template for each object pose and then obtain the complete template dictionary for different poses of each object category. The template consists of a set of Haar wavelet features, each of which is allowed to slightly perturb at different scales and orientations to account for the local structural deformation of each training sample. These active features are linearly combined to generate the template, or to fit to the image patch for the purpose of recognition.

Compared with previous works [3], [9], [21], [22], this multiview object representation has following characters.

a) Each deformable template can be efficiently learned via a small set of roughly aligned (in the same pose) positive samples without negative samples, using the shared sketch algorithm [34].
b) The proposed active Haar features can code the local structure variations of objects and suppress the disturbances of scene noises and similar background structure.
c) Combining with the spatio-temporal contextual information, we can roughly estimate the pose of object in scene to propose the candidate template, which is able to greatly reduce the computational cost in the matching step.

2) *Spatio-Temporal Context:* This is modeled independently as the prior knowledge for recognizing objects via both the current and the deferred observations in our unified inference.

a) Spatial context contains the viewing (camera) position and surface geometry. Note that we do not use the inter-object configuration as in many algorithms for object detection in images [10], [15], since objects usually move in a messy way and the inter-object configuration is not reliable enough in visual surveillance system. Actually, for each type of object, we have strong prior about the possible occurring locations and the object size at the specific location in the projected image plane, namely location-size constraint, which can be used to mitigate the false alarms for object recognition task. For example, with the fixed viewing position (camera has been roughly calibrated), the sizes of the objects in image plane can be estimated in a relative short range by applying the cross-ratio theorem. And the property of surface, like "ground," "sky," or "planar," also contributes to pruning false alarms. In order to obtain accurate spatial context model, we develop an annotation toolkit for viewpoint estimation and surface property modeling in an interactive fashion, where manual guidance is allowed to enhance the accuracy.

b) Temporal context is built up based on the pixel-level and instance-level consistency. Instead of observing a static frame, we cache a sequence of observations. For example, the temporal context at frame t is deduced on the $[t-\tau, t+\tau]$ frames, with τ as the parameter to define the period. For temporal context modeling, we assume that both pixels and instance remain consistent over a short period. Hence, at the pixel level, the intensity-variance on observed frames can be calculated by a background model [17], [32] (a constant probabilistic gray-scale distribution). Intuitively, one pixel in the current frame is proposed as foreground when it fits against the background model over a few frames. In this paper, we also develop a novel background modeling algorithm to obtain a more robust foreground estimation. At the instance level, a selected feature distribution is used to constrain moving objects, namely the current foreground object is labeled when it accepts the same label over frames. Therefore, we should first uncover the correspondence between the foreground regions in consecutive frames and then apply this constraint to help recognize the object category. This is mostly related to the traditional object tracking algorithms whereas the difference is that we only need recover the temporal correspondence within a short period of observed frames, instead of the entire object trajectory.

3) *Stochastic Sampling:* It is adopted for object recognition, which integrates the spatio-temporal contextual information with the deformable template matching. Within the Bayesian inference framework, we formulate the problem of object localization as a maximum-a-posteriori (MAP) problem, in which we take the spatio-temporal contextual knowledge as the prior term and deformable template matching as the likelihood term. In this paper, the Markov chain Mento Carlo (MCMC) sampler is used to search for the optimal solution by simulating the Markov chain in the overall solution space. Instead of exhaustive sampling, however, we use the spatio-temporal contextual information to reduce the computational ambiguities by narrowing the search space, as well as to drive the stochastic search in Markov chain effectively.

The key contribution of this stochastic sampling procedure is to design a set of reversible jumps and diffusions to simulate the Markov chain. The algorithm follows the data-driven MCMC principle [37], which has been proved to be able to rapidly obtain the nearly globally optimal solution.

The entire solution is summarized in Fig. 2. The main contributions of this paper include: 1) proposing a multiview category representation via deformable templates for object representation and detection; 2) presenting a unified and practical spatio-temporal contextual model for object recognition in visual surveillance systems; and 3) developing a flexible inference framework which integrates the above two components into a real-time object recognition system.

The remainder of this paper is organized as follows. We first introduce the object representation in Section II and then introduce the Bayesian formulation of object recognition task in Section III. Section IV details the inference procedure. In Section V, we discuss the implementation details on the spatio-temporal context model. Comparison experiments are presented in Section VI and the paper is concluded in Section VII with discussion on future work.

II. Multiview Object Representation via Deformable Templates

A. Active Haar Features

Wu *et al.* [34] provided the original theory and methodology for active basis model. The model is based on a linear representation using Gabor wavelet elements, which are localized, elongated, and oriented version of the original wavelet functions. Each Gabor element is like a stroke in the sketch of an object. In this paper, we replace the Gabor features with Haar features [22] and set the number of orientations as 8. Compared with the original method [34], the pursuit of Haar templates is less time-consuming while retaining similar performance. Fig. 3 shows the comparison of the templates learned by the original Gabor features and our variant with Haar features. Haar features are more computationally efficient than Gabor features because it can borrow the strength of integral image techniques in previous work [22].

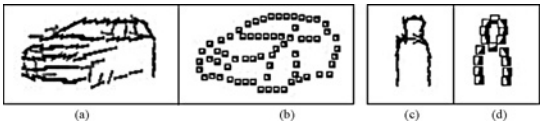

Fig. 3. Gabor features versus Haar features. (a), (c) Object templates using Gabor features. Each line stroke represents a Gabor basis. (b), (d) Object templates using Haar features.

Letting D denote the image lattice defined on the training image and $\mathcal{H}_{x,y}$ denote the Haar matrix centered at position (x, y), we can translate, rotate, and dilate $\mathcal{H}_{x,y}$ to obtain a general form of the Haar wavelets

$$B_{x',y',s,\alpha} = \mathcal{H}_{x''/s,y''/s}/s^2 \tag{1}$$

$$x'' = (x' - x)cos\alpha - (y' - y)sin\alpha \tag{2}$$

$$y'' = (y' - y)sin\alpha + (y' - y)cos\alpha \tag{3}$$

where the Haar feature $B_{x,y,s,\alpha}$ is centered at position (x, y), s is the scale parameter, and $\alpha \in \{k\pi/K, k = 0, ..., K-1\}$ (e.g., $K = 8$) is the orientation parameter. We normalize the Haar components to have zero mean and unit l_2 norm, and introduce local inhibitions such that they are orthogonal to each other. This constraint ensures that the strokes generated from the basis in the object template do not overlap with each other, and thus the selected elements are well connected to form a shape template.

For an image J, we can project it onto a Haar wavelet $B = B_{x,y,s,\alpha}$, and the projection coefficient is represented as $r =< J, B >= \sum_{u,v\in\wedge_B} J(u, v)B(u, v)$, where $\wedge_B$ indicates the image region covered by feature B and (u, v) denotes the coordinate within $\wedge_B$. Intuitively, if $| < J, B > |^2$ is the local maximum within a neighborhood window, there is an edge segment at (x, y) with orientation α and the size of the segment is related to the scale s. Thus, we can represent an image using a set of weighted Haar basis $\{B_i = B_{x,y,s,\alpha}, w_i; i = 1, 2, \ldots, n\}$ where w_i is the weight of the ith feature and n is the number of basis in the template. We further introduce a whitening transformation for each feature response, namely, $h(r) = threshold(r) = min(r, T_{whiten})$, where T_{whiten} is a threshold (e.g., $T_{whiten} = 16$). This operation ensures that heterogeneous features are well-calibrated and comparable, sharing the same distribution on natural image ensembles.

In order to code the structure variations of each training image $J_m, m = 1, \ldots, M$, we allow the selected base B_i to locally shift its location and orientation. Thus, letting w_i denote the feature weight, we can obtain a deformable template $\{B_i, w_i\}$ as

$$B_{x_i,y_i,s,\alpha_i} \approx B_{x_{m,i},y_{m,i},s,\alpha_{m,i}} \quad w_i = \sum_{m=1}^{M} r_{m,i}/M \tag{4}$$

if there exists $(\delta_{m,i}, \lambda_{m,i})$, and $x_{m,i} = x_i + \delta_{m,i}sin\alpha$, $y_{m,i} = y_i + \delta_{m,i}cos\alpha$, $\alpha_{m,i} = \alpha_i + \lambda_{m,i}$, where $\delta_{m,i} \in [-a, a]$, $\lambda_{m,i} \in [-b, b]$, and a and b are the bounds for the displacement in location and the turning in orientation (e.g., $a = 6$ and $b = \pi/K$), respectively. We call such element B_{x_i,y_i,s,α_i} as an *active Haar* and $\delta_{m,i}, \lambda_{m,i}$ as the activity of the element B_{x_i,y_i,s,α_i}. When deforming B_i to $B_{m,i}$, we have, $B_{m,i} = \arg\max_{B\approx B_i} | < J_m, B > |^2$, which subjects to the orthogonality constraint.

Algorithm 1. Shared sketch algorithm for template learning

- ***Input*** : Training samples $\{J_1, \ldots, J_M\}$;
- ***Output*** : Deformable template $\{B_i, w_i\}, i = 1, \ldots, n$;

1) For each $J_m \in \{J_1, \ldots, J_M\}$, and for every (x, y, α), compute feature response,
$$r_{m,x,y,s,\alpha} = h(| < J_m, B_{m,x,y,s,\alpha} > |^2);$$
Set $i \leftarrow 0$.
2) For each $J_m \in \{J_1, \ldots, J_M\}$, compute the local maximum value for each feature:
$$\widetilde{r}_{x,y,\alpha} = \max_{\delta\in[-a,a],\lambda\in[-b,b]} r_{m,x+\delta\sin\alpha,y+\delta\cos\alpha,s,\alpha+\lambda}.$$
Let $(\bar{\delta}, \bar{\lambda})$ be the (δ, λ) which achieves the maximum.
3) From the feature bank Ω_f,
   a) Choose the shared feature B_i that maximizes $\bar{r}_{x,y,s,\alpha} = \sum_{m=1}^{M} \widetilde{r}_{m,x,y,s,\alpha}/M$,
   b) Compute the perturbed version of each shared feature.
   For m=1 to M, let $\delta = \bar{\delta}_{m,i}$ and $\lambda = \bar{\lambda}_{m,i}$; calculate $B_{m,i}$ and its response $r_{m,i}$.
   Set $w_i = \sum_{m=1}^{M} r_{m,i}/M$.
4) Enforcing approximate non-overlapping constraint. For $m = 1$ to M, for each $B_{m,j}$, $j = 1, ..., n$, $j \neq i$, if $\langle B_{m,i}, B_{m,j}\rangle > \zeta$, then set $r_{m,x,y,s,\alpha}$=0. Here, $\langle\rangle$ returns the geometrical distance between two features and ζ is a threshold.
5) Stop if i = n, and normalize $\{w_i\}$ such that $\| w \|^2 = 1$. Otherwise let $i \leftarrow i + 1$, and go to (3).

B. Template Learning and Matching via Active Haar Features

Given a set of training samples in the same pose, denoted as $\{J_1, \ldots, J_M\}$, we adopt the shared sketch algorithm [34] to pursuit the Haar template $\{B_i, w_i\}$ from the complete feature bank Ω_f. Here, the samples are the image regions covered by foreground objects and are cropped from the training videos. The pursuit algorithm runs in a sequential fashion, and each step introduces a Haar feature to maximize the log-likelihood defined as follows:

$$\log \frac{p(J_m|B_i, w_i)}{q(J_m)} = \sum_{i=1}^{n} [\lambda_i h(r_{m,i}) - \log Z(\lambda_i)] \tag{5}$$

where $p(\cdot)$ indicates the foreground distribution over the responses of the selected basis and $q()$ indicates the reference distribution pooled from background structures. Herein, $\lambda_{m,i} = |w_{m,i}| = | < J_m, B_{m,i} > |^2$, and the normalizing constant is defined as $Z(\lambda) = E_q[exp\{\lambda h(r)\}]$. The score of (5) is a weighted sum of $h(r_{m,i})$. It evaluates the matching similarity between the image I_m and the deformed template. Here, we use the shared matching pursuit algorithm in [34]. In this parallel algorithm, each selected features B_i will be shared by all the training images, namely, for each m, a deformed feature $B_{m,i}$ is also selected to encode J_m. Algorithm 1 illustrates the details of this learning procedure.

Given the current input frame, we first apply the background model to propose the foreground regions, each of which is a candidate moving object. Then, we crop the foreground region

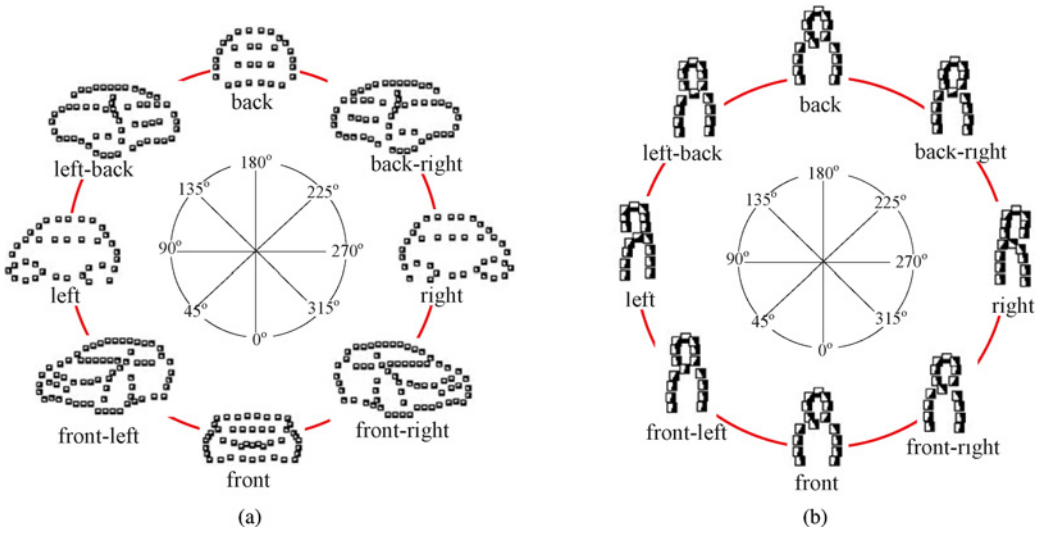

Fig. 4. Active Haar templates for the category of (a) cars and (b) pedestrians. The pose of one object in the image plane is usually determined by the tile angle and orientation angle. This figure shows the templates which are learned for the fixed title angle 45° and 8 orientation angles 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. (a) Haar templates of cars. (b) Haar templates of pedestrians.

Algorithm 2. Object category recognition algorithm using deformable template matching

- ***Input***: The testing image I; object template $\{B_i, w_i\}$;
- ***Output***: Matching score S;

1) For each (x, y), for all α's, compute the feature response, $r_{x,y,s,\alpha} = h(|< I, B_{x,y,s,\alpha} >|^2)$.
2) For each (x, y), for all α's, compute the local maximum response

$$\widetilde{r}_{x,y,\alpha} = \max_{\delta\in[-a,a],\lambda\in[-b,b]} r_{x+\delta\sin\alpha,\, y+\delta\cos\alpha,\, s,\, \alpha+\lambda}.$$

Let $(\bar{\delta}, \bar{\lambda})$ be the (δ, λ) which achieves the maximum, and denote the correspondent position as $(\bar{x}, \bar{y}) = (x+\bar{\delta}, y+\bar{\lambda})$.
3) For each Haar base $B_i = \{x_i, y_i, s_i, \alpha_i\}$, retrieve the local maximum response, $S_i = \widetilde{r}_{\bar{x}+x_i, \bar{y}+y_i, \alpha}$.
4) Compute the matching score, $S = \sum_{i=1}^{n} w_i \times S_i$.

as testing image and use the learned template to recognize its category. Letting I denote the cropped image, we assume the bounding box of object is centered at position $(x = 0, y = 0)$, and scan the object template over image I to fit the active Haar features into the image within the bounding box centered at each position (x, y). In each scanning step, we calculate the fitting response, namely the log-likelihood score defined in (5), and finally obtain the score map that measures the confidence of template matching. Algorithm 2 gives the details of this recognition algorithm. In implementation, we apply the above algorithm at multiple resolutions of the testing image, and choose the resolution that achieves the maximum score as the optimal one.

C. Multiview Template Dictionary

The proposed object model for each object category consists of a set of active Haar templates in different poses (views). In surveillance videos, the object pose is mainly determined by the orientation angle and tilt angle, namely the object view in frames. In order to create the complete multiview template dictionary, we quantize the angle space into linear bins, n_1 bins for 90° tilt range, and n_2 bins for 360° orientation range, to obtain totally $n_1 \times n_2$ templates of different views and then build the template dictionary Ω_l. For each pose, an object template is trained using the Algorithm 1 from a set of roughly aligned positive images.

It is worth noting that the choice of the angle number is essentially a tradeoff between performance and efficiency. Increasing the number of templates will reduce the intra-view varieties and thus boosts discriminating power of the template-based classifier, but at the same time increase the computational cost. In this paper, for each object category, we take one single tilt angle 45° and 8 orientation angles, including 0° (front), 45° (front-left), 90° (left), 135° (left-back), 180° (back), 225° (back-right), 270° (right), and 315° (right-front). Fig. 4 shows the learned template dictionaries for the categories of cars and pedestrians in images (a) and (b), respectively.

III. Contextual Modeling

In this section, we introduce the Bayesian formulation for object recognition with the spatial-temporal context modeling. Here, the objective is to compute the solution W based on the observed image I

$$W = \{K, R_0, O_i = \{V_i, R_i, L_i\}, i = 1, ..., K\} \tag{6}$$

where K denotes the object number in the current observed scene and O_i denotes the object instance characterized by three parameters: V_i (object view, or pose), R_i (instance patch), and L_i (the object label). Therein, R_0 denotes the background region.

Assuming that the object moves on the projected image plane, the observed object pose is mainly determined by the orientation angle and the vertical position. Given the scene camera calibration, the vertical position can be used to

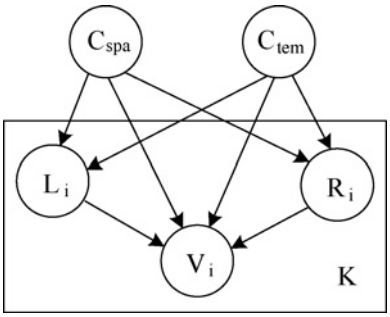

Fig. 5. Directed graphical model representing the joint probability distribution over the variables contained in the solution configuration W, as defined in (6). This model asserts that the variables $O_i = \{V_i, R_i, L_i\}$ are conditionally independent and identically distributed given variables C_{spa} and C_{tem}. The object number K is determined according to the current system time T.

compute the tilt angle between the ground plane and the line from the camera center to the object center. Thus, we represent the object pose V_i as

$$V_i = (\eta_{ori}, \eta_{tilt}) \tag{7}$$

where η_{ori} and η_{tilt} denote the orientation and the tilt angle, respectively. We also denote image region covered by the ith object as R_i

$$R_i = \{x_i, y_i, w_i, h_i, \Gamma_i = \partial R_i\} \tag{8}$$

where (x, y) is the center of object region in the image, w and h are the width and height of the bounding box, and ∂R is the shape region, which consists of the pixels with the same category label. Thus, the goal of this paper is to recognize the categories of foreground regions.

The object category label L_i, e.g., cars, bikes, and pedestrians, denotes the recognition result for the foreground region R_i. In our method, each category is represented by a set of deformable templates, namely, the template dictionary denoted as Ω_{L_i}. Thus, we can formulate the problem of object recognition as maximizing *a posteriori* within the Bayesian framework. Letting C_{spa} denote the observed evidence of the spatial/scene context, C_{tem} denote the evidence of temporal context and T denote the current system time T, the optimal solution W^* can then be solved as

$$\begin{aligned} W^* &= \arg\max_W P(W|I, C_{spa}, C_{tem}, T) \\ &= \arg\max_W P(I|W, C_{spa}, C_{tem}; \beta) P(W|C_{spa}, C_{tem}, T; \theta) \end{aligned}$$

where β and θ are the parameters for the likelihood and prior models, respectively. Fig. 5 shows the graphical model which illustrates the dependencies of the variables in Bayesian inference.

A. Prior Model

In our proposed framework, the spatial and temporal contexts are the key prior knowledge for inference. The former includes camera geometry and main surface property, and the latter determines the pixel-level and instance-level consistence. In this paper, we derive the prior model either using single type of contextual information or using both types. Thus, the prior model is defined as the joint product of the following terms:

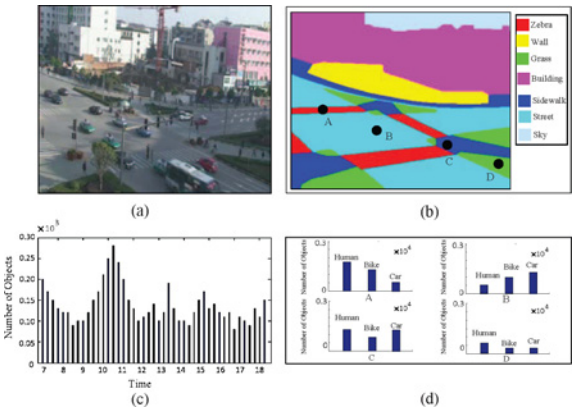

Fig. 6. Semantic scene modeling. (a) One observed frame captured in videos. (b) Major surfaces annotated by the interactive annotation tool, providing spatial contextual knowledge for scene understanding. (c) Vehicle density distribution over surfaces learned from the scene in image (a) by simply counting from 7:00 AM to 6:00 PM. (d) Object category density learned from the surfaces A, B, C, and D denoted in image (b). The distribution is conditional on surface semantic property. For example, on the zebra surface A, pedestrians occur in a relatively higher probability than the cars.

$$\begin{aligned} P(W|C_{spa}, C_{tem}, T; \theta) \propto P(K|T) \prod_{i=1}^{K} P(R_i, L_i|C_{spa}) \cdot \\ P(R_i, L_i|C_{tem}) P(V_i|R_i, L_i, C_{spa}, C_{tem}). \end{aligned} \tag{9}$$

Herein, the first term denotes the object density prior, the second one indicates the joint distribution of object location and object category/recognition given the observed spatial scene knowledge, the third term $P(R_i, L_i|C_{tem})$ uses the temporal contextual information for object recognition and the last one $P(V_i|\cdot)$ denotes the pose prior term, which utilizes both the spatial and temporal information. Besides, since it is usually difficult to directly sample the joint distribution in inference, we further decompose $P(R_i, L_i|C_{spa})$ as

$$P(R_i, L_i|C_{spa}) \propto P(L_i|R_i, C_{spa}) P(R_i|C_{spa}) \tag{10}$$

or

$$P(R_i, L_i|C_{spa}) \propto P(R_i|L_i, C_{spa}) P(L_i|C_{spa}). \tag{11}$$

We assume the two prior terms, namely $P(R_i|C_{spa})$ and $P(L_i|C_{spa})$, follow with the uniform distribution. Thus, by iteratively sampling from $P(L_i|R_i, C_{spa})$ and $P(R_i|L_i, C_{spa})$, we can approximately approach the distribution of $P(R_i, L_i|C_{spa})$. The first step is equivalent to predicting object density for each possible location in the imaging plane and the second step is equivalent to imposing the location-size constraint for the specific category of objects. In implementation, we set the iteration number empirically (e.g., five times) to make a tradeoff between the computation efficiency and the performance.

1) *Object Density Prior $P(K|T)$:* Object density prior $P(K|T)$ is defined on the distribution of object number over time, accounting for how busy the scene is. The distribution is discrete, and can be counted from the observed data directly

$$P(K|T) \propto hist_o(K|T). \tag{12}$$

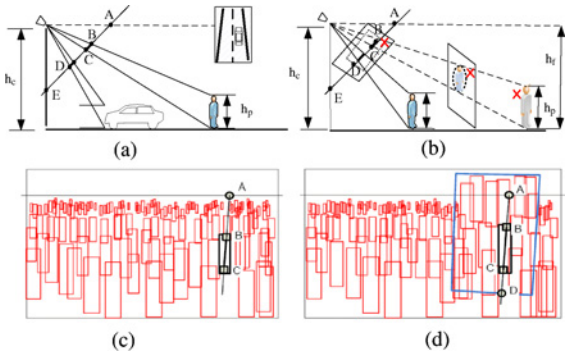

Fig. 7. Location-size constraint. (a) Without wall behind the object, the object size can be directly estimated by using the homograph between the imaging plane and the ground plane. (b) If wall exists, using only ground plane and imaging plane will result in seeing object behind the wall, which should be removed as false alarm. (c) Minimal blob size map which is pre-computed using (15) and used to provide the object size. (d) For the objects on the vertical surface, the homograph between vertical plane and imaging plane is used to suppress false alarms. The expected pedestrians height BC in (d) is larger than in (c).

In Fig. 6, the image (c) illustrates the object density histogram learned from the scene in image (a) based on the observations during the time from 7:00 AM to 6:00 PM.

2) *Recognition Prior* $P(L_i|R_i, C_{spa})$*:* We simply learn the object category density in videos and model the recognition prior as

$$P(L_i|R_i, C_{spa}) \quad \propto \quad hist_r(L_i) \tag{13}$$

where $hist_r(\cdot)$ denotes the learned histogram of category density. We first set the histogram $hist_r$ to an initial histogram at the beginning and then adapt it from time to time. Thus, this term becomes a Dirichlet distribution, which can be used as the prior model. The distribution of object type is surface-related, means that we distinguish the categories of local image regions and collect a histogram of object density over different categories. Fig. 6(d) shows the object type histograms learned from the scene in Fig. 6(a) for four different major surfaces. One can observe that, pedestrians appear on footpath with a high probability while on freeway with a low probability. Therefore, using the surface knowledge can further boost the discriminating power of the proposed model.

3) *Location-Size Prior* $P(R_i|L_i, C_{spa})$*:* In visual surveillance system, each type of objects, e.g., pedestrians, vehicles and bikes, has its own strong prior distribution about both the possible locations and the physical size at each location in observed images. For example, a pedestrian cannot be off the ground without the other support surfaces. Given camera calibration and ground-plane estimation, we can predict the expected physical size of each foreground blob in the image.

We model each type of object with a cuboid and make the following assumptions: 1) human beings can touch the ground plane, horizontal surfaces, or stair surfaces with bottom line, but can only touch vertical surface with side lines, and 2) vehicles and bikes can touch the ground plane, horizontal surfaces or stair surfaces with bottom line. Thus, the object size in the image plane can be directly estimated by projecting the cuboid on the touching surface.

Integrating scene geometry information, the location-size constraints can be defined as

$$P(R_i|L_i, C_{spa}) \propto P(x_i, y_i|L_i) \times P(h_i, w_i|x_i, y_i, L_i) \tag{14}$$

where the first term measures the location distribution of each category and the second term predicts the box size given the position and category label. Both terms can be calculated using a global 2-D map, which provides the minimal blob size at each location and orientation for each category [as illustrated in Fig. 7(c)]. The map can be pre-computed when system is initialized, and thus almost no additional burden is introduced in the inference stage.

Fig. 7 illustrates the calculation of the location-size constraint. Let B and C denote the pedestrian's head and feet, respectively, A denote the intersection of pedestrian and horizon line in the imaging plane, D denote the intersection of pedestrian and the vertical surfaces baseline, and E denote the vertical vanishing point. Also let h_p denote the height of pedestrian and h_c denote the height of camera. The expected size of an observed pedestrian on the ground-plane can be predicted as follows.

a) If there does not exist vertical surface behind the pedestrian, the human height BC can be predicted by (simply following the cross ratio theorem)

$$\frac{BC}{BA}/\frac{EC}{EA} = \frac{h_p}{h_p - h_c}. \tag{15}$$

b) If there exists vertical surface, the human height BC should be calculated by the following joint equations:

$$\begin{cases} \dfrac{BC}{BA}/\dfrac{EC}{EA} = \dfrac{h_p}{h_p - h_c} \\ \dfrac{AD}{AC}/\dfrac{ED}{EC} = \dfrac{h_c}{h_f} \end{cases} \tag{16}$$

where h_f is the vertical distance from the camera center to the human feet and can be canceled from above equations.

4) *Temporal Context Prior* $P(R_i, L_i|C_{tem})$*:* We define two temporal prior terms in our framework based on pixel-level and instance-level consistency over the deferred observations

$$P(R_i, L_i|C_{tem}) \propto P_{pix}(R_i, L_i|C_{tem}) \times P_{ins}(R_i, L_i|C_{tem}). \tag{17}$$

a) *Pixel-level consistency:* In our model, all the pixels in the current observed image I are proposed to be foreground (moving) region based on a learned background modeling (e.g., [17]), which is initially obtained and updated frame by frame. Letting θ_B be the model parameter, we denote the model as $L_{pix}(x; \theta_B)$, which describes how likely a pixel x belongs to background (further details are introduced in Section V). Thus, we use the following equation to model the pixel-level temporal prior:

$$\begin{aligned} P_{pix}(R_i, L_i|C_{tem}) \quad &\propto \quad P_{pix}(R_i|C_{tem}) \\ &\propto \quad \frac{1}{Z_{pix}} \times e^{-\sum_{x \in R_i} L_{pix}(x;\theta_B)} \end{aligned} \tag{18}$$

where Z_{pix} is the normalizing constant. We develop a novel background modeling method to robustly estimate the

Fig. 8. Object pose estimation by combining spatio-temporal information. With the camera viewpoint fixed, the pose of each object in the scene is mainly determined by two parameters: 1) orientation angle on the ground-plane, and 2) the angle between the ground plane and the line from the object center to the camera center. (a) Input frame. (b) Ground-plane projection of image (a) from reconstructed top view. The arrows represent the motion directions of the moving objects. Q denotes the camera center, G the object center, $\vec{GP}$ the predicted motion direction, and η_{ori} the orientation angle of the object.

background distribution for the challenging video clips (see Section V).

b) *Instance-level consistency:* It is assumed in this paper that an object instance should be consistent over time in both motion and appearance properties. We detect objects in each frame independently and make association between them in consecutive frames using the tracking algorithm (see Section V). Let R_t denote the current instance proposed, $\{R^{t-\tau}, \ldots, R^{t-1}\}$ denote the detection results in the previous frames within the sliding window of size τ (e.g., $\tau = 6$). We can thus maintain the following instance-level temporal context knowledge:

$$P_{ins}(R_i, L_i|C_{tem}) \propto \frac{1}{Z_{ins}} e^{-\sum_{k=1}^{\tau} L_{ins}(R_i^t, R_i^{t-k})} \tag{19}$$

$$L_{ins}(R_i^t, R_i^{t-k}) = \delta(L_i^t \neq L_i^{t-k}) \cdot \times D_f(F_{ins}(R_i^t), F_{ins}(R_i^{t-k})) \tag{20}$$

where Z_{ins} is the normalizing parameter, function $\delta(x)$ maps a real value variable x to $\{1, 0\}$, $F_{ins}(R)$ denotes the appearance features collected from region R, and the function $D_f(\cdot, \cdot)$ returns the similarity distance between two feature descriptors. In practice, we collect the local binary pattern (LBP) descriptor [19] as the appearance model, and have $D_f(\cdot, \cdot)$ defined as

$$D_f(F_1, F_2) = \frac{KL(F_1 \| F_2) + KL(F_2 \| F_1)}{2} \tag{21}$$

where $KL()$ denotes the Kullback-Leibler divergence.

5) *Pose Prior* $P(V_i|R_i, L_i, C_{spa}, C_{tem})$*:* We combine the spatial and temporal contextual information to introduce another type of prior over the object pose in scene. Object pose (view) is a very important cue to assist template selection when performing matching in the recognition step.

The prior distribution over object pose at each location can be defined as follows:

$$P(V_i|R_i, L_i, C_{spa}, C_{tem}) \propto hist_p(V_i|R_i, L_i) \tag{22}$$

where $hist_p$ is the 2-D histogram over object pose for the category L_i at location R_i. Given the object location R_i and category label L_i, the object pose $V_i = \{\eta_{tilt}^i, \eta_{ori}^i\}$ can be calculated by integrating the spatio-temporal contextual information.

In order to compute the tilt angle, we project the 2-D location of the observed blob onto the ground plane in image according to the estimated camera parameters. As illustrated in Fig. 8, we denote Q and G as the camera center and object center, and QG as the horizontal distance between the object and the camera in world coordinate (WC). Also, let h_c denote the height of camera. Both h_c and QG can be estimated from rough camera calibration. The tilt angle η_{tilt} can thus be calculated by

$$\eta_{tilt} = \arctan(h_c/QG). \tag{23}$$

For the orientation angle, we first trace the observed object frame by frame using the object correspondence method proposed (see Section V) to generate the 2-D trajectory, and then project the trajectory onto the ground-plane. Fig. 8(b) illustrates the constructed top view for the scene in Fig. 8(a). Letting P denote the point on the trajectory along the motion direction, the orientation angle η_{ori} can be directly calculated from the two vectors, $\vec{QG}$ and $\vec{GP}$.

B. Likelihood Model

In general, our contextual model can be combined with any window-based object detectors, which output the class-conditional log-likelihood ratio c. The likelihood term in our framework can be thus formulated as

$$P(I|W, C_{tem}, C_{spa}; \beta) \propto \prod_{i=1}^{K} P(I|O_i, C_{tem}, C_{spa}) \propto \prod_{i=1}^{K} e^{\lambda_{like} \cdot c_i} \tag{24}$$

$$c_i = \log \frac{P(I|L = l, R = r, V = v)}{P(I|L \neq l, R = r, V = v)} \tag{25}$$

where λ_{like} denotes the constant factor and l, r, v denote category label, bounding box, and object pose, respectively, as defined in Section III.

In this paper, we use the multiview object detector via deformable template as discussed in Section II. We first build a template dictionary with different poses for each object category to create a multiview classifier. Then, the object view or pose is heuristically selected by integrating the spatio-temporal contextual information. Finally, based on the template in the selected pose, we scan the testing window in the observed image I, and calculate the matching score, which includes both the feature fitting score and the global deformable energy, to compute the log-likelihood term defined in (25).

IV. Stochastic Inference

We present in this section a MCMC sampler to optimize the posterior probability as formulated in Section III. Benefiting from the contextual information, the valid solution space can be largely condensed. Thus, it is possible to apply the stochastic sampling algorithm for real-time recognition task. Compared with other energy minimization methods, such as graph-cut and belief propagation, MCMC [37] is the only known

general procedure to search for nearly globally optimal solution for a complex problem. To draw samples from the probability $P(W|I)$, it simulates a Markov chain which visits a sequence of states π_n in the solution space Ω_π.

The Markov chain $\mathcal{MC} =< \nu(\pi_0), \mathcal{K} >$ consists of an initial state $\nu(\pi_0)$ and one transition kernel $\mathcal{K}(\pi_A, \pi_B)$ which measures the conditional probability for moving from the state $\pi_A = P(W_A|I)$ to $\pi_B = P(W_B|I)$. A MCMC sampler is designed to explore the dynamics $\mathcal{K}(\pi_A, \pi_B) = Q(W_B \rightarrow W_A)$ on the chain with the acceptance probability

$$\alpha(\pi_A, \pi_B) = min\left(1, \frac{P(W_A|I) \times Q(W_B \rightarrow W_A)}{P(W_B|I) \times Q(W_A \rightarrow W_B)}\right). \quad (26)$$

In our framework, four reversible dynamics are designed as follows:

A. Object Addition

This move includes the foreground blob proposal R_q and category proposal L_q

$$Q_1(W_B \rightarrow W_A) = Q_{fore}(R_q|W_B, I_B) \times Q_{recog}(L_q|R_q, W_B, I_B) \quad (27)$$

$$Q_{fore}(R_q|W_B, I_B) = P_w \times P(R_q|W_B, I_B) \quad (28)$$

$$Q_{recog}(L_q|R_q, W_B, I_B) = P(L_q|R_q, I_B) \quad (29)$$

where P_w is the empirical probability for selecting an attention window in the image, $P(R_{fore}|\cdot)$ is the foreground blob proposal as defined in (14), and $P(L_q|\cdot)$ is the blob recognition proposal as defined in (13). Here, P_w is defined as the uniform distribution, i.e., $P_w = P_{unif}$.

B. Object Removal

It randomly selects and deletes one existing object O_q in the current solution

$$Q_2(W_B \rightarrow W_A) = P(O_q|W_B) \propto P_{unif}. \quad (30)$$

C. Object Type Change

It randomly selects one existing object O_q and assigns it to the label proposed by using the spatial context knowledge

$$Q_3(W_B \rightarrow W_A) = P(O_q|W_B) \times P(L_q|R_q, I) \quad (31)$$

where $P(L_q|\cdot)$ is the object type proposal as defined in (13).

D. Object Pose (Template) Change

This move changes the pose (template) for the randomly selected object O_q

$$Q_4(W_B \rightarrow W_A) = P(O_q|W_B) \times P(V_q|Q_q, I) \quad (32)$$

where $P(V_q|\cdot)$ denotes the object template proposal with the context defined in (22).

The first and second moves are referred to as jump dynamics while the third and fourth are referred to as diffusion dynamics, and thus the Markov chain is irreducible and aperiodic. Unlike the exhaustive Gibbs sampler, the above dynamics are integrated with the learned contexts for sampling proposal, inspirited by the data-driven MCMC principle in [37]. At the beginning of MCMC inference, we can obtain the initial state $\nu(\pi_0)$ as follows: 1) set K as the number of the foreground regions (subtracted by a background modeling module over the current frame) and R_i as the ith foreground region; 2) set R_0 as the background region; and 3) set L and R as the randomly selected values.

Fig. 9. Interactive viewpoint calibration in video.

V. Implementation

A. Spatial Context Modeling

The explored spatial context includes the camera viewpoint and surface information. We develop an interactive toolkit for spatial context modeling, which comprises two components: 1) viewpoint estimation in the scene, and 2) surface property estimation.

1) *Viewpoint Estimation:* As discussed in Section III, the camera (viewpoint) should be roughly estimated to obtain the object location-size and pose prior. As in previous literature [20] on camera calibration, the projection matrix P that connects the world coordinate (WC) with camera coordinate (CC) using five intrinsic parameters [focal length f_P, principal point(u_P, v_P), aspect ratio α_P and skew s_P] and six extrinsic parameters (X-Y-Z-translations and pan/orientation φ_{pan}, tilt φ_{tilt}, and roll φ_{roll}). In this paper, as illustrated in Fig. 7(a), the camera is fixed with only one degree of freedom, namely the height H_c, and the aspect ratio and skew rest assumed as $\alpha_P = 1$, and $s_P = 0$, respectively. The rest parameters (f_P, φ_{pan}, φ_{tilt}, φ_{roll}, H_c) can be solved by assigning the vanishing points and the principal point. The related theory and proof are detailed in [20].

Therefore, the viewpoint estimation problem is cast as the problem of accurately estimating the vanishing point (VP_X, VP_Y, VP_Z). Here we develop the toolkit for interactively calculating vanishing points from real input videos. We first manually select and label one moving object in the initial frame, and then label it in the following two frames. Since the objects in three different frames are in the same actual height,[1] the vanishing points (VP_X, VP_Y, VP_Z) can be calculated by manually labeled poles in a video, as illustrated in Fig. 9. In order to improve the accuracy, in practice, the calculation can be carried out based on a few moving objects simultaneously using embedded tracking function, inspired of the RANSAC principle [11].

2) *Surface Property Estimation:* Another piece of important prior knowledge for object recognition is the scene surface property. As analyzed in Section III, the object recognition prior is modeled as a learned distribution co-related with the surface evidence. In addition, after calibrating the viewpoint

[1]We assume the instances of the identical pedestrian in different frames have the same "actual height." Although different "actual heights" could be measured due to the oscillatory movement of the head, we obtain the satisfactory system performance from this roughly camera calibration.

Fig. 10. Sketch detection results refined by Gaussian filtering and edge linking. The primal sketch algorithm [6] is applied to obtain the sketchable regions (light color) and the unsketchable regions (dark color).

point, we can estimate the surface in the rough 3-D coordinates, as illustrated in Fig. 6(b). In this paper, we adopts the scene understanding method in [12] and [15] to develop an interactive annotation toolkit, which allows users to correct the results manually for enhancing both accuracy and efficiency.

B. Temporal Context Modeling

The temporal context includes two items: the pixel-level and instance-level consistency. The former is related with the background modeling and the latter is related with the classical object correspondence (tracking) task.

1) *Background Modeling using Image Primitives:* We develop a novel background modeling algorithm based on the information scaling theory proposed in [35]. The theory explains that the same object appearing at different scales produces image data with different statistical properties. In visual surveillance, objects appear at a wide range of scales in the images due to the change of viewing distance as well as the camera resolution. Accordingly, image patches with different scale have different properties and we decompose the observed image into two parts: sketchable region and unsketchable region. The former is mainly composed of blobs, end points, bars, junctions, corners and crosses of different degrees, and the later is composed of textured and flat regions. Formally, letting $\wedge$ denote the lattice defined on current image I, we have

$$\wedge = \wedge_{ske} \cup \wedge_{unske} \tag{33}$$

where $\wedge_{ske}$ denotes the patches within the sketchable region and $\wedge_{unske}$ denotes the patches within the unsketchable region. Here, the patches are collected from the image lattice $\wedge$ with a fixed size (e.g., 10×10 pixels). We apply the primal sketch algorithm [6] to obtain the sketchable regions. Fig. 10 shows an exemplar result of the sketch detection after refinement by Gaussian filtering and edge linking. The sketchable parts are shown in light color and the unsketchable parts are with dark color. Thus, each patch unit can be labeled as sketchable or unsketchable areas. We adopt different methods to describe each type of patch as follows.

First, we design a set of primitive prototypes filters to represent the sketchable patches and use the GMM of filter

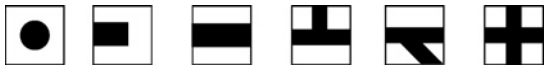

Fig. 11. Image primitives including blobs, end points, bars, junctions, corners and crosses of different degrees. Image primitives are composed of +1 (dark area) and −1 (light area), and allowed to rotate in 8 directions.

response values as the background model. Fig. 11 shows the six primitive prototypes used in this paper. For each patch unit, one of the six prototypes is selected as the filter by using the primal sketch algorithm [35]. The filter response is defined as the convolution of the specific primitive prototype [35] and the observed image region. In order to suppress scene noise and the disturbance of camera jitter, we also allow each primitive to locally shift its location to code the local structural variations. Let $B^{ske}_{\vec{x}}$ denote the primitive prototypes at position $\vec{x}_i \in \wedge_{ske}$, and $\wedge_{\vec{x}}$ indicate the region covered by $B^{ske}_{\vec{x}}$. Thus, the response of primitive $B^{ske}_{\vec{x}}$ can be calculated by

$$r_B(\vec{x}) = \max_{\vec{x}' \in \partial\vec{x}} \sum_{(u,v)\in\wedge_{\vec{x}'}} B^{ske}_{\vec{x}'}(u, v) \times I(u, v) \tag{34}$$

where $\partial\vec{x}$ denotes the neighborhood area of the position $\vec{x}$.

We assume the primitive response of each sketchable patch follow with the mixture of M_b Gaussian distributions, like the GMM background modeling method proposed in [5]. For a certain primitive $B_{\vec{x}}$, the probability of belonging to background region can be computed as

$$L_{ske}(\vec{x}; \theta^{ske}_k(\vec{x})) = \sum_{k=1}^{M_b} w_k(\vec{x}) \times \mathcal{N}(r_B(\vec{x}); \mu_k(\vec{x}), \sigma^{ske}_k(\vec{x})) \tag{35}$$

where $\mathcal{N}(\cdot)$ indicates the Normal distribution of the kth component, θ^{ske}_k indicates the model parameter which includes mean value μ_k and variance σ^{ske}_k, and w_k denotes the weight of the kth component. All above notations are parameterized by the location $\vec{x}$. For each position $\vec{x}$ in the tth frame, we first calculate the primitive response value r^t according to 34, and then update the Gaussian components that match r^t by the following equations:

$$w^t_k \approx (1 - \alpha^{ske}) \times w^{t-1}_k + \alpha^{ske} \times p_\delta(r^t; \theta^{ske}_k) \tag{36}$$

$$\mu^t_k \approx (1 - \alpha^{ske}) \times \mu^{t-1}_k + \rho^{ske} \times r^t \tag{37}$$

$$\sigma^t_k \approx (1 - \alpha^{ske}) \times \sigma^{t-1} + \rho^{ske} \times (r^t - \mu^t_k)^2 \tag{38}$$

$$\rho^{ske} = \alpha^{ske} \times \mathcal{N}(r^t; \mu^{t-1}_k, \sigma^{t-1}_k) \tag{39}$$

where α^{ske} is a forgotten parameter. The term $p_\delta(r^t; \theta^{ske}_k)$ is defined as a delta function, namely, if θ^{ske}_k is the first matched Gaussian component, $p_\delta(r^t; \theta^{ske}_k) = 1$; otherwise, 0. If none of the M_b components match the primitive response r^t, the least probable component is replaced by a distribution with the value r^t as its mean, an initially high variance, and a low weight parameter.

Second, we represent the unsketchable patches using the local binary pattern descriptor (LBP) [19] and also describe each patch using GMM model. Letting $H_{\vec{x}}$ denote the binary histogram computed from the image patch centered at $\vec{x} \in \wedge_{unske}$, we can define the background model for the unsketchable area using a single Gaussian distribution as

$$L_{unske}(\vec{x}; \theta^{unske}) = \mathcal{N}(H(\vec{x}); H_m(\vec{x}), \Sigma^{unske}(\vec{x})) \tag{40}$$

where the parameter θ^{unske} includes mean histogram H_m and the covariance matrix Σ^{unske}. For each position $\vec{x}$ in the tth frame, we collect the LBP descriptor H^t, and use it to update the learned Gaussian distribution $\mathcal{N}(;H_m, \Sigma^{unske})$ by the following equations:

$$H_m^t = (1-\alpha^{unske}) \times H_m^{t-1} + \rho^{unske} \times H^t \tag{41}$$

$$\Sigma^t = (1-\alpha^{unske}) \times \Sigma^{t-1} + \alpha^{unske} \times (H^t - H_m^{t-1}) \times (H^t - H_m^{t-1})^T \tag{42}$$

$$\rho^{unske} = \alpha^{unske} \times \mathcal{N}(H^t; H_m^{t-1}, \Sigma^{t-1}) \tag{43}$$

where α^{unske} is the forgotten factor.

In summary, for current observed video sequence, we define the background model $L_{pix}(\vec{x};\theta^B)$ as

$$L_{pix}(\vec{x};\theta_B) = \begin{cases} \sum_{k=1}^{M_b} w_k(\vec{x})\mathcal{N}(r_B(\vec{x}); \mu_k(\vec{x}), \sigma_k^{ske}(\vec{x})), \vec{x} \in \wedge_{ske} \\ \mathcal{N}(H(\vec{x}); H_m(\vec{x}), \Sigma^{unske}(\vec{x})) \quad \vec{x} \in \wedge_{unske}. \end{cases} \tag{44}$$

2) *Object Correspondence:* In order to obtain the correspondence between the moving blobs (foreground region) in consecutive frames, we use the tracking algorithm in [27], which proposes to perform mean-shift with scale-space kernel to optimize for blob location and scale. Then, the correspondence is used as the instance-level temporal prior as defined in Section III. Note that we use the correspondence as the initial proposal, and this way of using the algorithm [27] makes our approach less vulnerable to the quality of this tracking step.

VI. Experiments

We evaluate the proposed framework in four aspects: 1) deformable template learning; 2) background modeling; 3) benefit verification for contextual information; and 4) object category recognition as well as localization. The average system speed is around 7–10 frames/s on a Pentium-IV 2.2 GHZ computer without code optimization in the C++ platform.

We use two public datasets to evaluate the algorithmic effectiveness. The first one is the LHI Dataset [4]. We choose 9 video clips of different scenes, each of which is of about 10 minutes with the frame rate of 25 frames/s and the frame size of 352×288 pixels. The bounding boxes of foreground objects are manually annotated for each frame. The second one is the PETs dataset, from which we select 3 video clips and manually annotate the foreground objects. Thus, we finally obtain a total of 12 videos. For each video clip, we use the first 1000 frames for training and the remaining frames for testing. These videos provide challenging scenes with heavy occlusions, scale changes or complex background structure. In order to create the positive training samples for building multiview template dictionaries, we also manually crop 60 object patches in the same pose for each category from the above videos.

A. Experiment I: Deformable Template Learning Using Active Haar Features

This experiment illustrates the results from the shared sketch algorithm based on active Haar features. In all tests, the threshold of whitening operation takes $T_{whiten} = 16$. The size of Haar features is set as 8×8 pixels. (x, y) is sub-sampled every 2 pixels in both horizontal and vertical directions. The orthogonality tolerance of the Haar elements for local inhibition is set as $\zeta = 0.1$. The shift along the normal direction is set as $\delta_{m,i} \in [-a, a] = [-6, 6]$ pixels. The shift of orientation is set as $\lambda_{m,i} \in [-b, b] = \{-1, 0, 1\}$ angles out of $K = 8$ angles. There are two tuning parameters in our experiments. One is the number of elements, n, which is usually different for different category: 60 for bikes, 80 for cars, and 20 for pedestrians. The other parameter is the resolution of the training images, which is also different for each category: 110×90 pixels for bikes, 120×90 pixels for cars and 60×80 pixels for pedestrians.

We apply the sketch algorithm to a set of $M = 60$ roughly training aligned images in the specific pose for each category. In Fig. 12, we sketch and plot the Haar templates $\{B_i; i = 1, \ldots, n\}$ in the first column, and shows the corresponding training samples in other columns. The intensity of the bar

Fig. 12. Learned deformable templates using active Haar features. The first plot in each row shows the learned object template via active Haar features and the rest plots show the training samples. Each Haar element is represented by a bar. These elements can locally perturb their locations and orientations, such that the template becomes deformable.

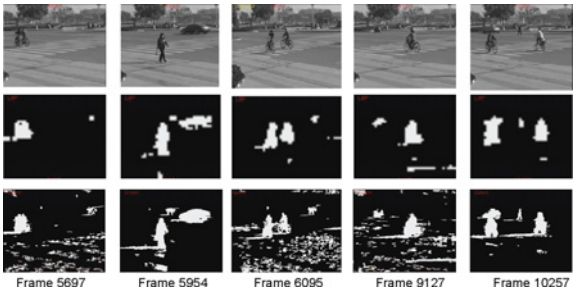

Fig. 13. Comparison of the background modeling results. We plot the foreground mask images generated by the standard GMM [5] and our proposed method. Top row: original images; Middle row: background subtraction results of our method; Bottom row: background subtraction results of GMM.

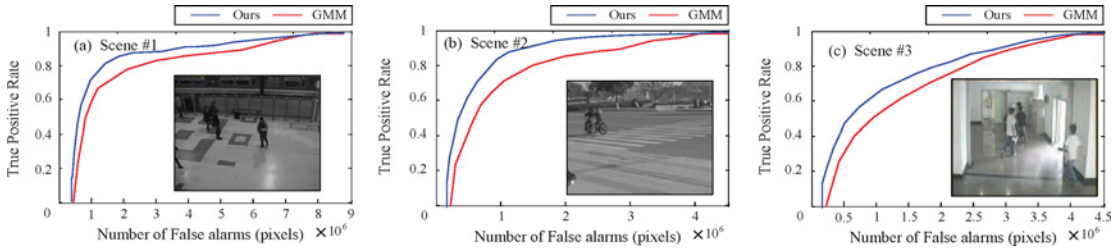

Fig. 14. ROC curve comparison for background modeling between GMM [5] and our method. Experiments are conducted on three surveillance videos. (a) Scene #1. (b) Scene #2. (c) Scene #3.

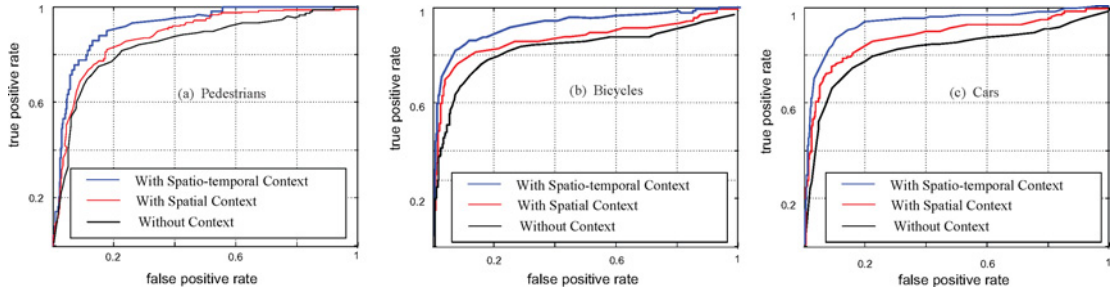

Fig. 15. Object recognition with different priors for the categories of (a) pedestrians, (b) bikes, and (c) cars. Black curves represent the recognition accuracies without context information, red curves represent the recognition accuracies using spatial context only, and blue curves denote the results by using integrated spatio-temporal contexts.

B_i is the average value $\sum_{m=1}^{M} h(B_{m,i})/M$. It is worth noting that for the second sample of the bikes (the third row in Fig. 12), the strong edges in the background are not sketched, because these edges are not shared by other examples and hence ignored. Fig. 4 shows more object templates learned for multiview object representation.

B. Experiment II: Background Modeling for Observed Videos

In this experiment, we evaluate the proposed background modeling method. The learning rate α (for both sketchable and unsketchable regions) is set as 0.005 and the number of Gaussian components for modeling sketchable region is set as $M_b = 16$. The patch size takes 10×10 pixels. For comparison, we implement the popular GMM [5] as the baseline and set its parameters the same as in [5]. Fig. 13 shows some mask images generated by these two algorithms. From the results, we can observe that GMM produces more false alarms than our method, especially when there exist reflection, shadows, or intersection of moving objects in scenes.

We also quantitatively compare these two algorithms and show the ROC curves of foreground detection results in Fig. 14. In this test, we use three different challenging scenes, in which the areas of sketchable regions are 52.1%, 44.5% and 47.5%, respectively. Our method achieves higher detection accuracy compared to the GMM algorithm over all the three scenes.

C. Experiment III: Benefit Verification for Contextual Information

In order to quantitatively analyze the improvement brought by different context components, we run the proposed method three times using different contextual information: 1) without context information, namely we use the uniform distribution for all the prior model terms defined in Section III-A; 2) with spatial context, including the object density prior term [see (12)] and recognition prior term [see (13)]; and 3) with spatio-temporal context. We perform the evaluations on the 12 video clips and plot the ROC curves for the categories of pedestrians, cars, and bikes in Fig. 15. Each black curve represents the recognition results without context information, each red curve represents the recognition results using spatial context only, and each blue curve represents the recognition results for using the integrated spatio-temporal context. From these results, we can observe that: 1) for all the categories, both the spatial and temporal contextual information contribute a lot, and 2) vehicles are reasonably influenced by the temporal context more than other two categories, since they move in a more predictable way. Besides, we also observe that, the more complex the scene is (e.g., with frequent occlusion or lighting changing), the more important role the context plays.

D. Experiment IV: Object Category Recognition and Localization

We test our method for the task of object detection and segmentation. In this experiment, we use the template dictionaries learned in Experiment I, which includes $n_1 \times n_2 = 1 \times 8 = 8$ templates for each category. We run the recognition algorithm on 8 resolutions of the observed image. The scaling factor is from 0.5 to 1.2. Fig. 4 shows the template dictionaries for the categories of cars and pedestrians, and Fig. 16 shows 10 representative scenes with the recognition results. Although there exist large intra-class variations, heavily occluded objects, and viewpoint/scale variations in the video clips, the performance of our proposed method is encouraging.

We compare our approach with the popular algorithm proposed by Viola *et al.* in [23], which describes a pedestrian detection system by integrating intensity information

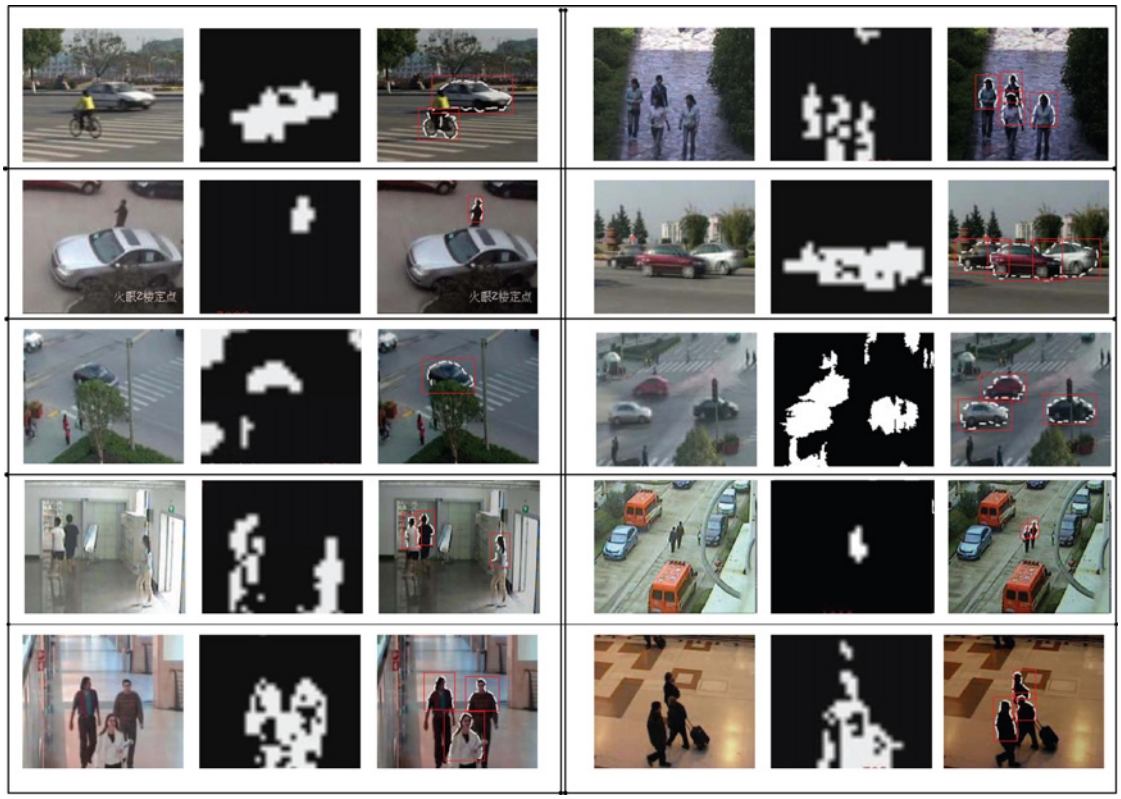

Fig. 16. Some representative results on selected videos. Each result shows: the observed scene (left), foreground mask (middle), and matching verification (right).

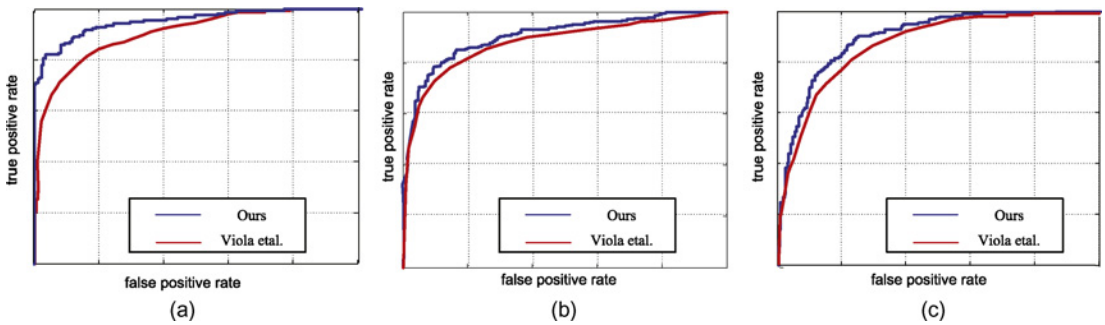

Fig. 17. Comparison ROC curve results with [23] for the categories of (a) pedestrians, (b) bikes, and (c) cars.

with motion information. Also, we apply their algorithm for detecting other object categories including cars and bikes. In order to train the classifier for each category, we collect a set of 450 positive examples and 450 negative examples. Each positive training example is a pair of patches cropped from two consecutive frames that contains the specific category of object. Negative samples are similar patch-pairs which do not contain the specific objects. Here, the patch size is set as the same as in Experiment I for each category. We keep other parameter settings the same as in [23]. Fig. 17 shows the evaluating ROC curves for different categories, where the red and blue curves denote the results of our method and the algorithm [23], respectively. These results show that our method outperforms the algorithm in [23] for all tests. However, the improvement for the pedestrian detection is higher than that for car detection. A possible explanation is that the intra-class variance of the cars is higher than that for pedestrians, and it is predictable that the detection accuracy can be further improved by introducing more templates.

We also show the multi-class object recognition results with comparison to TextonBoost [15]. In these tests, we consider 3 types of objects, i.e., cars, pedestrians and bikes. For labeling images, the method of TextonBoost constructs a discriminative model by exploiting three types of information: appearance, shape and local spatial configuration. Object classification and feature selection are achieved by using shared boosting to give an efficient classifier which can also be applied to a large number of classes. For comparison, the TextonBoost classifier of each category is obtained using 60 positive samples and 500 negatives, which are manually cropped from the training video sequence. We set the parameters as the same as in [15]. Fig. 18 illustrates the confusion matrix, in which the

	Cars	Bikes	Pedestrians
Cars	91.83% 88.17%	3.83% 4.17%	4.34% 6.66%
Bikes	4.70% 8.50%	89.60% 79.40%	5.70% 12.10%
Pedestrians	3.90% 4.40%	5.50% 7.00%	90.60% 88.60%

Fig. 18. Confusion matrix with percentages row-normalized.We compare our method (in red) and TextonBoost [15] (in black). Our method achieved a higher overall accuracy of 90.67% compared to TextBoost algorithm of 85.35%.

red number denotes the result from our method and the black number for that from TextonBoost. The overall average accuracy of our method is 90.67% and that of TextonBoost is 85.35%. Therefore, our method achieves higher accuracy and less false alarms compared to TextonBoost, by benefiting from the unified spatio-temporal context information combined in the inference process.

VII. Conclusion and Future Work

In order to detect and localize objects in visual surveillance, this paper presented a flexible framework that combines the spatio-temporal contextual information with a novel deformable template matching procedure. In the proposed method, various spatio-temporal cues were explored for the top-down verification and the solution was achieved by the stochastic MCMC method. Comparison experiments showed that our proposed method outperformed those state-of-the-art algorithms.

In this paper, the context modeling was scene-specific and thus not flexible enough for moveable visual surveillance system. We are planning to further study this problem in our future work.

Acknowledgment

The authors would like to thank Prof. Y. Wu for providing the source code for active basis learning, and thank Prof. S.-C. Zhu for extensive discussions. The data used in this paper were provided by the Lotus Hill Annotation Project [4].

References

[1] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. 235–241.

[2] A. Torralba, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, May 2007.

[3] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[4] B. Yao, X. Yang, and S. Zhu, "Introduction to a large scale general purpose groundtruth dataset: Methodology, annotation tool, and benchmarks," in *Proc. Energy Minization Method Comput. Vis. Pattern Recognit.*, LNCS 4697. 2007, pp. 169–183.

[5] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, vol. 2. 1999, pp. 2246–2254.

[6] C. Guo, S. Zhu, and Y. Wu, "Primal sketch: Integrating texture and structure," *Comput. Vision Image Understanding*, vol. 106, no. 1, pp. 5–19, 2007.

[7] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[8] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 10–17.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 2. Oct. 2006, pp. 2137–2144.

[11] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in *Proc. Int. Conf. Comput. Vis.*, vol. 1. 2003, pp. 199–206.

[12] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. Int. Conf. Comput. Version*, vol. 1. 2005, pp. 654–661.

[13] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[14] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am.*, vol. 2, pp. 1160–1169, Jul. 1985.

[15] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint appearance, shape and context modeling for multiclass object recognition and segmentation," in *Proc. Eur. Conf. Comput. Vision*, vol. 1. 2006, pp. 1–15.

[16] K. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: A graphical model for recognizing scenes and objects," in *Proc. Neural Inform. Process. Syst. Conf.*, 2003.

[17] L. Li, W. Huang, Y. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[18] L. Wolf and S. Bileschi, "A critical view of context," *Int. J. Comput. Vision*, vol. 69, no. 2, pp. 251–261, 2006.

[19] M. Heikkil and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[20] N. Krahnstoever and P. Mendonca, "Bayesian autocalibration for surveillance," in *Proc. Int. Conf. Comput. Vision*, vol. 2. 2005, pp. 1858–1865.

[21] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vision*, vol. 2. 2006, pp. 428–441.

[22] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[23] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[24] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, pp. 55–79, Jan. 2005.

[25] R. Collins, A. Lipton, T. Kanade, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa, "A system for video surveillance and monitoring," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-12, May 2000.

[26] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jul. 2003, pp. 264–271.

[27] R. Collins, "Mean-shift blob tracking through scale space," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 1. Jul. 2003, pp. 235–241.

[28] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[29] S. Avidan, "SpatialBoost: Adding spatial reasoning to AdaBoost," in *Proc. Eur. Conf. Comput. Vision*, vol. 4. 2006, pp. 386–396.

[30] S. Ullman, E. Sali, and M. Vidal-Naquet, "A fragment-based approach to object representation and classification," in *Proc. Int. Workshop Visual Form*, 2001, pp. 1–4.

[31] S. 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.

[32] W. Hu, H. Gong, S. Zhu, and Y. Wang, "An integrated background model for video surveillance based on primal sketch and 3-D scene geometry,"

in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2008, pp. 1–3.
[33] W. Zhang, B. Yu, G. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with multiple features," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 2. Jul. 2005, pp. 323–330.
[34] Y. Wu, Z. Si, C. Fleming, and S. Zhu, "Deformable template as active basis," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.
[35] Y. Wu, S. Zhu, and C. Guo, "From information scaling of natural images to regimes of statistical models," *Quart. Appl. Math.*, vol. 66, no. 1, pp. 81–122, 2007.
[36] Z. Tu, "Probabilistic boosting tree: Learning discriminative models for classification, recognition and clustering," in *Proc. Int. Conf. Comput. Vision*, vol. 2. 2005, pp. 1589–1596.
[37] Z. Tu and S. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.

Xiaobai Liu has been pursuing the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, since September 2006.

Since December 2008, he has been a Research Associate, under Prof. S. Yan, with the Learning and Vision Group, National University of Singapore, Singapore. He spent one year as a Research Associate with the Lotus Hill Institute, Wuhan, under the supervision of Professor S.-C. Zhu from 2007 to 2008. He has published more than 12 articles over a series of research topics. His research interests include computer vision, machine learning, and large scale image retrieval.

Liang Lin received the B.S. and Ph.D. degrees from the Beijing Institute of Technology (BIT), Beijing, China, in 1999 and 2008, respectively. He was a joint Ph.D Student at the Statistics Department, University of California, Los Angeles (UCLA), from 2006 to 2007.

He was a Post-Doctoral Research Fellow with the Center for Image and Vision Science, UCLA, as well as a Senior Research Scientist with the Lotus Hill Research Institute, Wuhan, China, from 2007 to 2009. He is currently an Associate Professor with the School of Software, Sun Yat-Sen University, Guangzhou, China. He has published more than 30 academic papers. His current research interests include but not limited to computer vision, pattern recognition, computer graphics, and virtual reality.

Dr. Lin has received a number of honors, including several scholarships in his Ph.D. study, Beijing Excellent Students Awards in 2007, the Excellent Ph.D. Thesis Award of BIT in 2008, and the Best Paper Runners-Up Award in NPAR 2010, as well as others.

Shuicheng Yan (M'06–SM'09) received the Ph.D. degree from the School of Mathematical Sciences, Peking University, Beijing, China, in 2004.

He spent three years as a Post-Doctoral Fellow with the Chinese University of Hong Kong, Shatin, Hong Kong, and then with the University of Illinois at Urbana-Champaign, Urbana. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He has authored or co-authored over 140 technical papers over a wide range of research topics. In recent years, his research interests have focused on computer vision (biometrics, surveillance, and internet vision), multimedia (video event analysis, image annotation, and media search), machine learning (feature extraction, sparsity/non-negativity analysis, and large-scale machine learning), and medical image analysis.

Dr. Yan has served on the editorial board of the *International Journal of Computer Mathematics*, has served as a Guest Editor of a special issue for *Pattern Recognition Letters*, and has been serving as the Guest Editor of a special issue for *Computer Vision and Image Understanding*. He has served as a Co-Chair of the IEEE International Workshop on Video-Oriented Object and Event Classification (VOEC'09) held in conjunction with ICCV'09. He was the Special Session Chair of the Pacific-Rim Symposium on Image and Video Technology in 2010. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Hai Jin (M'98–SM'06) received the Ph.D. degree in computer engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1994.

He is currently a Professor of Computer Science and Engineering with HUST. He is the Dean of the School of Computer Science and Technology, HUST. He was with the University of Hong Kong, Shatin, Hong Kong, from 1998 to 2000, and was a Visiting Scholar with the University of Southern California, Los Angeles, from 1999 to 2000. He is the Chief Scientist with ChinaGrid, the largest grid computing project in China. He has co-authored 15 books and published over 400 research papers. His current research interests include computer architecture, virtualization technology, cluster computing and grid computing, peer-to-peer computing, network storage, and network security.

Dr. Jin received the German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz, Chemnitz, Germany, in 1996. He received the Excellent Youth Award from the National Science Foundation of China in 2001. He is a member of the Association for Computing Machinery (ACM) and the Grid Forum Steering Group. He is the Steering Committee Chair of the International Conference on Grid and Pervasive Computing and the Asia-Pacific Services Computing Conference. He is a member of the Steering Committee of the IEEE/ACM International Symposium on Cluster Computing and the Grid, the IFIP International Conference on Network and Parallel Computing, the International Conference on Grid and Cooperative Computing, the International Conference on Autonomic and Trusted Computing, and the International Conference on Ubiquitous Intelligence and Computing.

Wenbing Tao received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004.

Since 2005, he has been with the School of Computer Science and Technology, HUST, where he is currently an Associate Professor. He was a Research Fellow with the Division of Mathematical Sciences, Nanyang Technological University, Singapore, from March 2008 to March 2009. He has published numerous journal and conference papers in the area of image processing and object recognition. His current research interests include the area of computer vision, image segmentation, object recognition and tracking, image search engines, and multimedia retrieval.

Dr. Tao serves as a Reviewer for many journals, such as the *International Journal of Computer Vision*, IEEE TRANSACTIONS ON IMAGE PROCESSING, *Pattern Recognition, Image Vision Computing*, *Pattern Recognition Letters*, and others.