Layered Graphical Models for Tracking Partially Occluded Objects

Vitaly Ablavsky, *Member*, *IEEE*, and Stan Sclaroff, *Senior Member*, *IEEE*

Abstract—We propose a representation for scenes containing relocatable objects that can cause partial occlusions of people in a camera's field of view. In many practical applications, relocatable objects tend to appear often; therefore, models for them can be learned offline and stored in a database. We formulate an occluder-centric representation, called a graphical model layer, where a person's motion in the ground plane is defined as a first-order Markov process on activity zones, while image evidence is aggregated in 2D observation regions that are depth-ordered with respect to the occlusion mask of the relocatable object. We represent real-world scenes as a composition of depth-ordered, interacting graphical model layers, and account for image evidence in a way that handles mutual overlap of the observation regions and their occlusions by the relocatable objects. These layers interact: Proximate ground-plane zones of different model instances are linked to allow a person to move between the layers, and image evidence is shared between the observation regions of these models. We demonstrate our formulation in tracking pedestrians in the vicinity of parked vehicles. Our results compare favorably with a sprite-learning algorithm, with a pedestrian tracker based on deformable contours, and with pedestrian detectors.

Index Terms—Computer vision, image representation, tracking, graphical models.

1 Introduction

Tracking multiple targets using fixed cameras with nonoverlapping views is a challenging problem. One of the challenges is predicting and tracking through occlusions caused by other targets or by fixed objects in the scene. Considerable effort has been devoted toward developing appearance models that are robust to partial occlusions [60], [30], [3] and toward developing tracking algorithms that can cope with a short-term loss of observations [67], [41], [62]. A complementary line of research has focused on learning static occlusion maps using large sets of observations accumulated over time [40].

In this paper, we consider scenarios where it is impossible to learn a static occlusion map. This is often the case when the scene consists of both people and large objects whose position is not permanently fixed. These objects may enter, leave, or relocate within the scene during a short time span. We call such objects *relocatable objects* or *relocatable occluders*.

Scenarios that include relocatable occluders are quite common. Fig. 1 shows four examples. In each scenario, relocatable objects tend to cause severe occlusions of people in the scene and, since these objects are movable, learning a single fixed occlusion map is impractical. For instance, in the supermarket scenario, shoppers accumulate items in grocery carts. The imaging setup typically consists of a ceiling-mounted camera that looks along the aisles. Because of the camera's shallow depression angle, people and shopping carts frequently occlude each other. In the parking lot surveillance example, fixed cameras with nonoverlapping views survey a parking lot with multiple parked vehicles. It is often the case that the cameras are mounted at a shallow depression angle to allow for wide coverage. This tends to lead to frequent occlusions of pedestrians by vehicles in each camera view. And as the distance from the camera increases, occlusions become more severe, while the apparent size of pedestrians gets smaller.

In each of the above scenarios, the person-tracking system must contend with numerous relocatable occluders in the scene and their adverse impact on image observations. Therefore, in scenarios such as parking lot surveillance, the 3D model-based trackers of [35], [8] are likely to be distracted by intervehicle occlusions. Furthermore, the image resolution typical in such scenarios makes it difficult to apply 3D alignment techniques based on high-contrast edges [28]. We advocate an approach that decomposes the problem into dynamic scene-maintenance and, conditioned on a scene, tracking a variable number of people. To make our approach practical, we propose an *implicit* 3D representation which can be rapidly assembled online via a database lookup of probabilistic graphical model layers corresponding to the relocatable occluders. In this layer-based formulation, localization of a relocatable occluder's mask may be possible via simple image-based approaches such as template-matching, even when low image contrast and resolution preclude the use of richer image-based models [30].

In many practical applications, relocatable objects are of known classes and tend to be observed repeatedly over time. Because many examples of relocatable occluders are observed it is possible to learn a function that decides a relocatable occluder's class. Furthermore, these objects' 3D models can be acquired using standard methods. Because the cameras are fixed, 2D image masks of relocatable

• *The authors are with the Department of Computer Science, 111 Cummington St., Boston University, Boston, MA 02215.*
E-mail: {ablavsky, sclaroff}@cs.bu.edu.

Manuscript received 18 Aug. 2009; revised 4 May 2010; accepted 24 Dec. 2010; published online 1 Mar. 2011.
Recommended for acceptance by V. Pavlovic.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-08-0546.
Digital Object Identifier no. 10.1109/TPAMI.2011.43.

Fig. 1. This figure shows examples of people moving around relocatable occluders such as (a) cars, (b) shopping carts, (c) magazine racks on wheels. In (d), an example of fixed occluders, desks and workstations, is shown.

occluders can be precomputed from their 3D models and stored in a database. In some applications, it may be advantageous to further subdivide relocatable occluders into distinct subclasses, and compute subclass-specific 3D models and their 2D image masks.

In our representation, a scene is modeled as a composition of depth-ordered layers of probabilistic graphical models. The number of these models equals the number of relocatable occluders in the scene at any given time. Each graphical model is comprised of an occlusion mask, a set of image observation regions for observing a person's motion near and around this occlusion mask, and a first-order Markov model for the person's motion around the relocatable object. The person's motion model is defined in the relocatable object's object-centered coordinate system, but this motion model is then mapped into the image plane where observations are obtained. Individual models are then composed to yield a coherent observation and state space.

We demonstrate our formulation in a parking lot surveillance application. First, we propose an approach to account for the image evidence that is sensitive to the number, position, and depth-order of pedestrians moving on this discrete state space. Next, using Viterbi optimization, we show how a variable number of pedestrians can be tracked in a sliding temporal window fashion. Because the state space is vehicle-centric, we not only estimate positions of pedestrians in the image plane, but also motion patterns around vehicles in the ground plane; yet the ground plane is never explicitly referred to during computations.

In summary, we make the following contributions:

- We develop a *representation* [1] for scenes containing relocatable occluders. Specifically, the scene is a composition of depth-ordered layers of graphical models. These models can be composed on-the-fly to form a layered global scene model.
- We propose a *solution* to a specific problem that makes use of this new representation: tracking of pedestrians in a parking lot crowded with parked vehicles.

We also note what the paper is *not* about. This paper is not about a new appearance descriptor for person-tracking. In fact, in our example application, due to the very small apparent size of people and severity of occlusions, we employ binary images generated by background subtraction. This paper is not about learning a static occlusion map. Methods for learning such maps [40], [63], [16] are complementary to our approach. This paper is not about free-space tracking, for which off-the-shelf algorithms of [49], [47], [12] can be applied. This paper is not about high-level activity recognition; the output of the algorithm is a sequence of estimates of vehicle and pedestrian locations. However, the mapping of pedestrian estimates from the image plane to locations around parked vehicles would provide valuable cues to an activity-recognition system.

We assume that the only source of information is a single fixed camera or a set of fixed cameras with nonoverlapping views. This is the case in many, but not all scenarios. If multiple overlapping views are available, occlusions must still be accounted for in individual views. However, it may be advantageous to track directly in 3D. The applicability of layers of graphical models to these multiview scenarios is a promising direction for future research, but it is not within the scope of this paper.

2 Related Work

In this section, we first review related *representations* for dynamic scene analysis, and then discuss related *approaches* to tracking persons and vehicles.

2.1 Representations of Dynamic Scenes

The W^4 system [15] demonstrated tracking and activity analysis of pedestrians walking in isolation or in groups. A pedestrian was tracked as a bounding box and, in order to preserve her identity after occlusions, a temporal texture template was maintained. To reason about her activities, body parts were inferred from her bounding contour. It is not clear how the boundary analysis performed in cases of reduced resolution or low contrast. Because the W^4 system did not infer depth-order of overlapping pedestrians, it may have had difficulties with prolonged occlusions. It is also not clear how the system would handle static occluders in the scene.

To extract and depth-order multiple moving regions from uncalibrated video, layered representations have been proposed [57], [17]. In [20], layers with probabilistic occupancy that could undergo nonrigid deformations between frames were investigated. Layered representations with stronger segmentation priors were proposed in [19], [50], [25]. With the exception of [50], which focused on tracking vehicles from an airborne platform so that targets appeared small relative to the image frame and did not overlap each other, layer extraction algorithms tend to be computationally expensive. Sequential layer-tracking and layer-learning was proposed in [51], but the method might still be impractical for a real-time system.

One limitation of these layered representations is their inability to model the motion of targets around layers. Intuitively, a representation that provides stronger motion priors, such as the likely motion of a pedestrian in the vicinity of a parked vehicle, might enable a target tracker to cope with prolonged occlusions. In [66], a scene was modeled as a stack of interleaved foreground and background layers, allowing the algorithm to track objects through occlusions. However, this approach still did not address the motion-around-layers aspect of scene modeling.

The use of layered models in tracking systems can be computationally demanding; therefore, methods that exploit domain knowledge about appearance changes have been proposed that offer real-time performance. In [39], a ground-plane to image-plane mapping was learned from the bounding boxes of pedestrians and vehicles; the projected height of pedestrians and vehicles as a function of their ground-plane positions was also learned and this function was used to track targets in image coordinates. In [40], this method was extended to cope with static occluders, such as the subway turnstiles. However, multiple trajectories of pedestrians had to be observed to learn these static occluders over time. Therefore, this method might not be practical to apply to scenes comprised of relocatable occluders. An approach to handling static occluders was presented in [56], but it relied on extracting occluding boundaries of the foreground objects; this may not work well under reduced image resolution and increased sensor noise.

When a dynamic scene is comprised of objects of known types, e.g., vehicles, it may be practical to acquire their 3D models offline. The scene can then be represented by instances of these models whose pose varies over time. Such model-based tracking has been applied to vehicles [35], [8] and pedestrians [29], [26]. Although a vehicle's pose may be tracked more accurately with a 3D model than with only a 2D bounding box, such methods tend to be sensitive to abrupt changes in the target's appearance, such as those caused by relocatable occluders.

When it is not feasible to acquire 3D target models or when the application does not require an estimate of the target's position in every frame, a volumetric representation may be appropriate. In [42], [54], voxel-carving of a bounded 3D volume seen from multiple calibrated views was demonstrated, while in [36], probabilistic voxel occupancy for change detection was proposed. Such methods can be computationally intensive if the desired voxel resolution is high and the number of views is large. Depending on the application, the computed volumetric representation may need further parsing to extract individual targets of interest.

2.2 Tracking of People

Methods in this section are grouped by the granularity of the representation. A common approach to tracking a person is with a monolithic representation—a 2D bounding box [15], [48] or an ellipse [7] if tracking in the image coordinates, or a 3D bounding box [12] if tracking in the ground plane. To maintain the identities of targets, a region descriptor may be added, such as a temporal texture template in [15], a color histogram in [12], or region covariance in [37]. However, in some multiview approaches [22], [23], appearance descriptors were considered unreliable and were omitted. A shortcoming of such monolithic representations, particularly when employed in a single-camera system, is that they may not be adequate to estimate the targets' depth-order or to accurately localize targets during abrupt and severe occlusions.

To address the shortcomings of monolithic representations, various forms of partitioning a persons' model into subregions have been studied. A method for tracking and depth-ordering a variable number of closely spaced people using a single calibrated camera was presented in [18]. A person was modeled as a generalized cylinder, and its color appearance in the image plane was modeled as a grid of uniformly spaced disks. In [21], each foreground blob was partitioned into regions in a polar coordinate system and the color distribution of each region was then estimated. In the multiview approach of [33], a person was modeled as a cylinder partitioned into horizontal slices, and for each slice a separate appearance model was maintained. Although a fixed model partitioning may lead to better performance than a monolithic model, it may be nontrivial to design a partitioning that anticipates all possible variations in a target's appearance.

In order to better cope with interperson occlusions or to meet the requirements of a specific application, methods that align a part-based model to each tracked person have been proposed. In [47], which tracked pedestrians passing by a store-window display to determine their focus of visual attention, the target model was comprised of a texture-based face component and a 2D bounding box covering the rest of the body; no depth-ordering was required by the application. Depth-order and segmentation of people in close proximity was the main objective in [10], where a person was modeled in the image plane as an ellipse that was partitioned based on image evidence into horizontal slices corresponding to head, torso, and legs regions. In [27], [26], the pedestrian's local appearance was modeled via a codebook, and the pedestrian's shape defined implicitly via the spatial probability distribution over the codebook's entries. Methods that combine discriminatively trained body-part detectors in a tracking framework have been proposed in [60], [62], [65]. A method to combine tracking and detection with a generative part-based model was proposed in [2]. While part-based models offer a principled way to handle interperson occlusions, they may not deal well with abrupt, severe occlusions caused by relocatable objects in a scene. Moreover, given the richness of the part-based representations, these methods require sufficient pixel resolution.

2.3 Relation to the Proposed Approach

Although the scene representations mentioned earlier have been successfully used in practical applications, these representations are not immediately applicable to relocatable occluders. When there is no overlap between multiple camera views and when occlusions in each nonoverlapping view are severe, maintaining an accurate 3D scene model may become a challenge. In such cases, it might still be possible to maintain a 2D layered representation of the scene. However, previous works on layered representations did not model the motion of a person around these layers. An additional shortcoming of the layered representations mentioned in this section is the unique challenge posed by

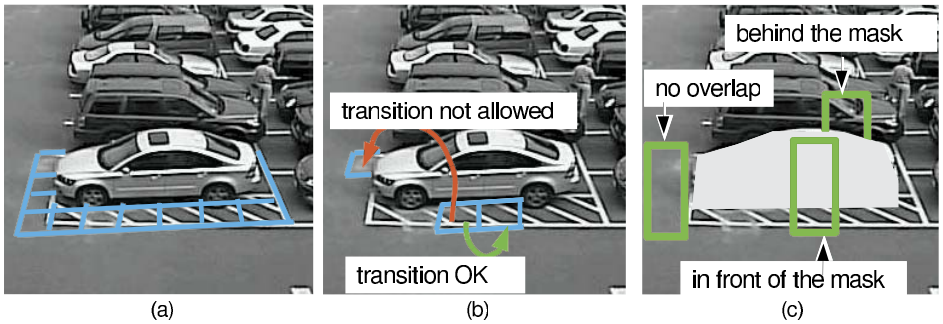

Fig. 2. To convey the basic idea we focus on one application, tracking of pedestrians in a parking lot, and define our object-centered representation to the white sedan at the front of the parking lot. (a) We tessellate the ground plane around the vehicle into *activity zones*. (b) A first-order Markov process on these activity zones captures motion patterns of pedestrians around vehicles (c) In the image plane, rectangular *observation regions*, three of which are shown, are depth-ordered with respect to the vehicle's *occlusion mask*. This object-centered representation, called a *graphical model layer*, is instantiated for every parked vehicle and composed as depth-ordered layers that interact.

the relocatable objects: There might not be enough time to learn the occlusion masks of these objects before they come to rest or while they are at rest.

Our approach addresses these shortcomings. A graphical model layer representation encapsulates our knowledge of how a person moves around a relocatable object. Multiple instantiated layers interact, yielding a global model for the likely motion patterns of people moving in the vicinity of these layers. Because occlusion masks of relocatable objects are acquired offline and stored into a database, a global scene model is assembled on-the-fly rather than learned from scratch every time a relocatable object enters the field of view.

A preliminary version of this work appeared in [1]. The main contributions over [1] include:

- a new, more general formulation is presented,
- experiments are conducted with scene models that have been updated automatically,
- qualitative comparisons are presented with [44] and [51], and
- detailed quantitative results are reported.

An additional contribution is in formulating a Viterbi sliding-window pedestrian-tracking algorithm. A Reversible-Jump Markov Chain Monte Carlo (RJ MCMC) recursive filtering algorithm was formulated in a previous version of our framework [1]. The key advantage of the RJ MCMC formulation in [1] is that it models uncertainty in the positions of targets; this uncertainty is represented as samples in the Markov Chain at each time step. However, we have found that, in practice, the specifics of the RJ MCMC death move and severe occlusions tend to yield a set of samples that is too diffuse to provide a definitive answer to the question: Where is each person in the scene? The Viterbi-based tracker proposed in this paper computes a point estimate of each person's location.

3 Approach

We begin by conveying the basic idea of our approach using one practical application: parking lot surveillance. We pick this example for the sake of concreteness and not because our representation favors this particular application over other examples mentioned earlier.

The key concept behind the approach is an occluder-centered representation. This representation encapsulates our prior knowledge of a person's motion around a relocatable object in the ground plane and where this motion would be observed in the image plane. For illustration, we define our object-centric model for the white sedan at the front of the parking lot in Fig. 2.

Because only the projection of motion in activity zones may be observed, our occluder-centric model is instantiated as a graphical model layer in the image plane. Expressing 3D mobility and visibility constraints implicitly in a layered framework opens the possibility of employing simpler image-plane techniques during inference.

In our global scene representation, a separate graphical model layer is instantiated for every parked vehicle. For example, in Fig. 2, a layer is instantiated for the white sedan, the black SUV behind it, etc. Overlapping layers are depth-ordered, so we refer to the global scene representation as *depth-ordered layers of graphical models*.

An application of our formulation to pedestrian-tracking in a parking lot is summarized in the diagram of Fig. 3. Input video frames feed into a module that tracks vehicles as they arrive, park, or depart. When a vehicle parks, a precomputed object-centric graphical model is retrieved from the database of such models and instantiated as a layer in our global scene representation. Precomputing a database of models is possible since the camera is fixed and a number of methods can be used to obtain ground-plane calibration [39], [31]. When a vehicle "un-parks," its layer is removed from the global scene representation. Layers in the global scene representation interact: Observations are shared between the instantiated models and links are added so that a pedestrian can transition between layers. By accounting for image evidence in a way that is sensitive to the number, image location, and depth-ordering of the pedestrians near vehicles, the system tracks a variable number of such pedestrians over time. We next present our approach in depth.

3.1 Local Representation: Graphical Model Layer

A graphical model layer encapsulates our prior knowledge of a person's motion around a relocatable object in the ground plane and our knowledge of where this motion would be observed in the image plane. This is an object-centered representation.

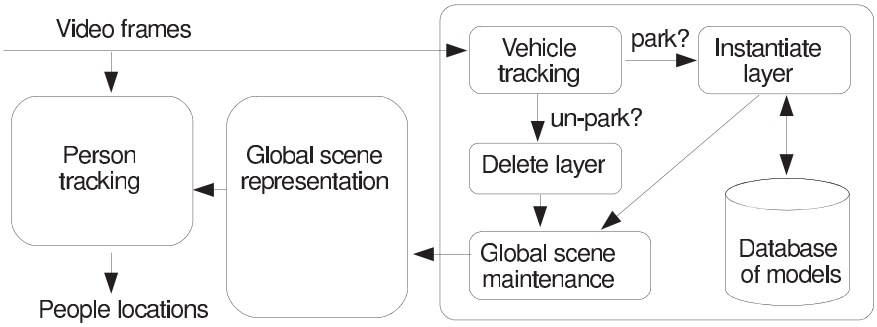

Fig. 3. This figure shows an application of our formulation: tracking pedestrians in a parking lot. The proposed scene representation enables tracking of pedestrians despite prolonged, severe occlusions; this representation is assembled on-the-fly using a database of precomputed graphical model layers. Please see the text for further details.

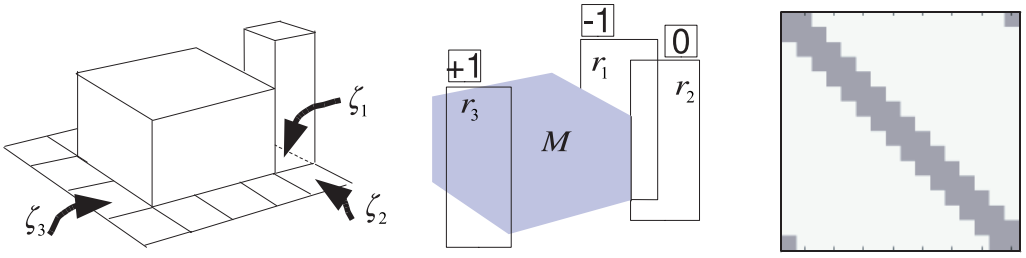

Fig. 4. (a) A relocatable object in 3D is shown as a squat box, while a person standing behind it is shown as a tall box. A subset of the ground plane around a relocatable occluder is partitioned into activity zones ζ_i. A person's motion on these zones is modeled as a first-order Markov process. For example, a stochastic transition from ζ_1 to ζ_2 is likely, while a transition from ζ_1 to ζ_3 is not. (b) The occlusion mask M and a subset of R comprised of three depth-ordered observation regions are shown. (c) The transition matrix for a model with 16 activity zones encodes the ring topology where the self-transition and transitions to the left and the right neighbors are equally likely; darker colors encode higher probability of transition.

TABLE 1
Notation for Graphical Model Layer Formulation

ζ	ground-plane *activity zone*
Y	a person's state defined over ζ's; y_t is her location at time t
M	image-plane occlusion mask of the graphical-model layer
r	image-plane observation region depth-ordered with respect to M
R	the set of all observation regions for this graphical-model layer
O	the set of per-pixel binary occlusion variables in R
Z	observations in R
L	the number of instantiated graphical-model layers in our global scene representation
D	depth-order of the instantiated graphical-model layers

Our notation for the graphical model layer formulation is summarized in Table 1. We partition a subset of the ground plane around the relocatable object into N bounded regions, called *activity zones*. In our implementation, activity zones are equally sized nonoverlapping squares, where each square is large enough to accommodate a person. In Fig. 4a, the squat box corresponds to a relocatable object. Three of its activity zones are labeled $\zeta_1, \zeta_2, \zeta_3$. We emphasize that this is not the only way activity zones can be defined. For example, one could contemplate scenarios in which zones vary in size, overlap, or do not even lie in the same plane.

We are ultimately interested in person-tracking on activity zones. Therefore, we encode the person's state as a random binary N-dimensional vector Y. For Y to be a valid state, its elements must sum to one.

We model Y_t as a first-order Markov process, where $p(Y_{t+1}|Y_t)$ reflects dynamics of the person and mobility constraints imposed by the relocatable object. In the example shown in Fig. 4a, it is reasonable to expect that the probability of a transition from ζ_1 to ζ_2 is high, while the probability of a transition from ζ_1 to ζ_3 is very low. These transition probabilities are summarized in a transition matrix; a transition matrix for a simple example model is illustrated in Fig. 4c.

A person standing in ζ is approximated in the image plane by a bounding rectangle r, called an *observation region*. In our implementation, observation regions correspond to a height of 1.8 meters to fully cover people of likely heights. Projection of the relocatable object yields an *occlusion mask* M. Observation regions are depth-ordered with respect to M. Fig. 4b shows the occlusion mask corresponding to the relocatable object in Fig. 4a. For the three activity zones $\zeta_1, \zeta_2, \zeta_3$, we show the corresponding observation regions r_1, r_2, r_3 and their depth-order with respect to M. Observation region r_1 is marked with -1, indicating that it is behind M, r_2 is marked with 0, indicating that this observation

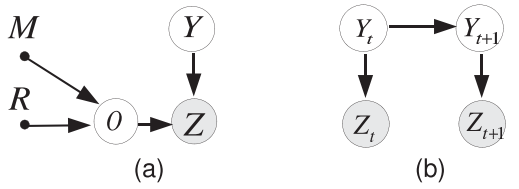

Fig. 5. (a) A single graphical model layer is a generative model for the observations Z given the occlusion mask M, depth-ordered observation regions R, occlusion variables O, and the location of a person Y. When this graphical model layer is instantiated into the scene representation, R and M are determined, and the probability of occlusion O is computed. Therefore, during inference, we only need to estimate the person's position Y given the image evidence Z. (b) The generative model for a single person moving around a relocatable occluder and resulting image evidence is summarized by a two-slice Hidden Markov Model; here the dependence on R, M, and O is implicit.

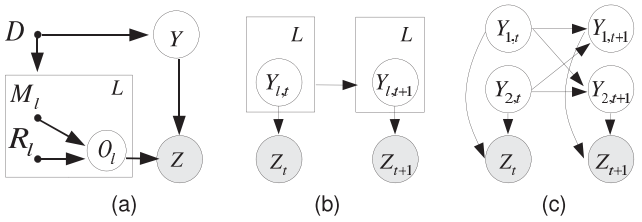

Fig. 6. (a) In our global scene representation, comprised of L layers, observations Z are generated by conditioning on a person's location in the global state space Y and occlusion variables in all O_ls. Occlusion variables and the global state space are a function of the depth-order D that is determined when the layers are instantiated. Therefore, during inference, our objective is to infer Y given Z. (b) A two-slice DBN makes the structure of Y explicit as a collection of individual Y_ls. Connections between $Y_{l,t}$ and $Y_{l,t+1}$ are determined by the depth-order D. (c) An example scene with two graphical model layers. A person moves between zones around a relocatable object, but may also transition between these objects as specified by $p(Y_{1,t+1}|Y_{1,t}, Y_{2,t})$ and $p(Y_{2,t+1}|Y_{1,t}, Y_{2,t})$.

region does not intersect M, and r_3 is marked with $+1$, indicating that it is in front of M. The set of depth-ordered observation regions is denoted by R.

We represent M as a set of binary random variables and their probabilities of being equal to one. For every depth-ordered observation region r_i and every pixel $u \in r_i$, we define a binary random occlusion variable $o_{i,u}$. Intuitively, $p(o_{i,u})$, the probability of occlusion, can be computed from the observation region's depth-order and the occupancy of the occlusion mask at that pixel. We define $O = \{o_{i,u}\}$ to be the set of all occlusion variables in all observation regions.

In many practical applications, image evidence is computed for every image location u. Example pixel level features include dense optical flow, frame difference, color likelihood, etc. We denote by z_u an observation at a pixel u, and let $Z = \{z_u\}$ for all pixels in the image. These observations are generated by conditioning on a particular state of a person y and occlusion variables O.

Our graphical model layer with dependencies between all of its variables made explicit is summarized in Fig. 5a. A Hidden Markov Model interpretation is given in Fig. 5b, where only the image evidence and activity zone nodes are shown.

3.2 Global Scene Model: Depth-Ordered Layers of Graphical Models

Our global scene representation is instantiated by specifying the model type, location, scale, and orientation for each of the L graphical model layers in the scene. The number of layers varies over time, as different relocatable objects arrive in and depart from the camera's field of view.

The graphical model layers are arranged according to the depth-order D. In our implementation, we represent D as a set of variables, one for each pair of layers. The value of each variable is $+1$ if the first layer occludes the second layer, -1 if the second layer occludes the first one, and 0 if the two layers do not interact. We add a constraint on these variables to ensure that no two layers may simultaneously occlude each other.

A person's state space in this layered representation is defined on a concatenation $Y_1, \ldots, Y_L$, with a constraint that any realization $y_1, \ldots, y_L$ sums to one. When not concerned with the internal structure of the state space, we will refer to it as Y.

The graphical model corresponding to L instantiated graphical model layers is shown in Fig. 6a. The generative model for a single person moving around the relocatable occluders is summarized by Dynamic Bayes Net (DBN) in Fig. 6b, where dependence on D is not drawn to avoid cluttering the diagram. As an example, in Fig. 6c, we consider a case where $L = 2$ and transitions from Y_2 to Y_1 and from Y_1 to Y_2 are allowed. This DBN encodes the following scene model: A person either transitions within Y_1, i.e., in the activity zones around the first relocatable occluder, or within Y_2, i.e., in the activity zones around second relocatable occluder. There exists an edge between a zone of the first relocatable occluder and the second relocatable occluder that allows the person to transition between these occluders. Note that the structure of this example DBN is not learned, but is instead determined by the interaction of the instantiated layers. This interaction yields a global transition graph, which we discuss next.

For a pair of zones owned by the same instantiated layer, the allowed one-step transitions are defined by that layer's graphical model. For a pair of zones owned by different layers, a one-step transition may be allowed if these zones are proximal and not separated by an occlusion mask. This intuition can be formalized by the following *connectivity test*: Two vertices from different layers are linked by an edge if all of the following conditions are satisfied: 1) Their observation regions overlap and are approximately the same size and 2) these observation regions are not separated by an occlusion mask.

In the example in Fig. 7a, three graphical model layers with masks M_1, M_2, and M_3 are shown, with arrows between masks indicating their depth-order. Observation region r_1 is behind M_1, r_2 and r_3, are, respectively, in front of and nonoverlapping M_2, and r_4 is nonoverlapping M_3. Given this configuration of observation regions and occlusion masks, our connectivity test is satisfied for the pairs of zones (ζ_1, ζ_2) and (ζ_3, ζ_4). The added edges in the graphical model for the scene are shown as straight lines in Fig. 7b. In Fig. 7c, we show instantiated occluder masks for a subset of real scene. The resulting global transition matrix is shown in Fig. 7d.

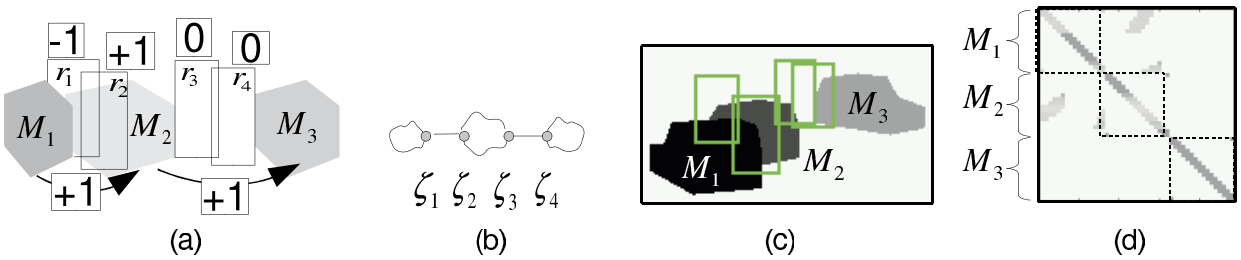

Fig. 7. (a) An example global scene model with three depth-ordered, interacting layers. (b) Based on layer membership, overlap, relative size, and depth-order of observation regions r_1, r_2, r_3, r_4, our *connectivity test* is satisfied for zone pairs (ζ_1, ζ_2) and (ζ_3, ζ_4). The edges added to the global transition graph are shown as straight lines. (c) A subset of a global scene from the PETS 2001 data set, described in Section 4, with M_1, M_2, M_3 corresponding to the rightmost three models of the bottom-left image in Fig. 8; four observation regions are shown. (d) The global transition matrix for this scene contains transition matrices from each graphical model layer on its block-diagonal; these matrices are outlined with dashed lines. Layers that own masks M_1 and M_2 interact, and the likely transitions between their activity zones that satisfy our connectivity test can be seen off the block-diagonal.

3.3 Accounting for Image Evidence in the Depth-Ordered Layers of Graphical Models

Given the global scene model and incoming video, we want to account for image evidence Z in each frame as a function of a person's location Y. For the sake of demonstrating our approach, we consider the case of binary features $z_u \in \{0, 1\}$. These features may be obtained by a moving-pixel detection algorithm based on background subtraction. Background subtraction tends to work on video sequences with relatively low resolution and contrast, as the recent approaches of [12], [13] demonstrate. Other features may be possible within our model, but this is sufficient to demonstrate the proposed method.

We summarize our notation for accounting for image evidence in Table 2. Recall that the occlusions in each observation region r are modeled by a set of binary random variables $\{o_u\}$, $u \in r$. Let $\tilde{o}_u$ be shorthand for $p(o_u = 1)$ and $\tilde{m}_{l,u}$ be shorthand that pixel u belongs to the occlusion mask of graphical model layer l. The probability that a pixel u in the observation region r is occluded can be computed from the set F_r of layers in front of r's layer:

$$\begin{aligned}\tilde{o}_u &= 1 - p(o_u = 0) = 1 - \prod_{l \in F_r} p(m_{l,u} = 0) \\ &= 1 - \prod_{l \in F_r} (1 - \tilde{m}_{l,u}),\end{aligned} \tag{1}$$

which follows since the event that a pixel u is not occluded means that it is not covered by any mask from the set F_r.

We define the mask of a person in an activity zone to be a rectangle equal to the corresponding observation region. Formally, $p(s_u = 1|y)$ equals one for any $u \in r_y$ and zero everywhere else.

Zero or one-person case. Given a person in state $Y = y$ and the corresponding observation region r_y, the probability of image evidence at pixel u is

$$\begin{aligned}p(z_u|y, R, M) &= \sum_{s_u} \sum_{o_u} p(z_u|s_u, o_u, y, R, M) p(s_u, o_u|y, R, M) \\ &= \sum_{s_u} \sum_{o_u} p(z_u|s_u, o_u) p(s_u|y) p(o_u|R, M) \\ &= \sum_{o_u} p(z_u|s_u = 1, o_u) p(o_u|R, M),\end{aligned} \tag{2}$$

which sums over all possible assignments of $s_u \in \{0, 1\}$ and $o_u \in \{0, 1\}$ in the first line, applies the chain rule on the second line, and substitutes the person's mask in the third line. Then,

$$p(z_u = 1|y, R, M) = q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u) \tag{3}$$

and

$$\begin{aligned}p(z_u|y, R, M) = &[q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u)]^{z_u} \\ &\times [1 - (q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u))]^{1 - z_u}.\end{aligned} \tag{4}$$

We assume that, conditioned on the moving person in r, individual pixels are uncorrelated:

$$p(z_r|y, R, M) = \prod_{u \in r} p(z_u|y, R, M). \tag{5}$$

For any pixel $u \notin r_y$, we have

$$p(z_u|R, M) = (q_2)^{z_u} (1 - q_2)^{1 - z_u}. \tag{6}$$

TABLE 2
Notation for Accounting for Image Evidence

$m_{l,u}$	probability of the mask occupancy of layer l at image location u
$o_{i,u}$	probability of occlusion of observation region i at image location u
$s_{k,u}$	probability of the k-th person mask occupancy at image location u
y	activity zone corresponding to the location of one person
$\mathbf{y}$	activity zones $y_1, \ldots, y_K$ for K persons
r_y	observation region corresponding to the activity zone y
q_1	probability that a pixel of a moving object is assigned the moving label
q_2	probability that a pixel belonging to the stationary background is assigned the moving label

TABLE 3
Notation for Tracking Formulation

T	number of video frames in a temporal window; $t \in [1,\ T]$
$y_{1:T}$	a sequence of activity zones, i.e., a person's track
$\mathbf{Y}_t$	location of all the people at time t
$\mathbf{Y}_{1:T}$	sequence $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$
$\mathcal{L}(y_{1:T}; \mathbf{Y}_{1:T}, Z_{1:T})$	log-likelihood of a track given other tracks $\mathbf{Y}_{1:T}$ and observation sequence $Z_{1:T}$

For any pixel outside of the union of all the observation regions, we have $p(z_u) = 0.5$. These three disjoint regions account for all the pixels in the image.

Multiple-person case. It is straightforward to extend our formulation to the case of $K \geq 1$ people occupying distinct activity zones $y_1, \ldots, y_K$. Although in some applications, activity zones may be designed to accommodate multiple people, this case is left for future work.

Our generative model accounts for the dynamics of K people and their occlusion relations in the observation regions. If D^{persons} specifies the depth-order and u is an image pixel that belongs to the nonempty intersection of the observation regions corresponding to $\mathbf{y} = (y_1, \ldots, y_K)$, we can write

$$p\big(z_u|\mathbf{y}, D^{\text{persons}}_{y_1,\ldots,y_K}; R, M\big) = p(z_u|\mathbf{y}; R, M). \tag{7}$$

For the purpose of demonstrating our framework, we have assumed that the image evidence takes the form of binary image masks. Binary image masks do not convey information about the depth-order. Therefore, if at least one $o_{u,k} \neq 1$,

$$p(z_u = 0|s_{u,1} = 1, \ldots, s_{u,K} = 1, o_{u,1}, \ldots, o_{u,K}) = 1 - q_1, \tag{8}$$

and it can be shown that

$$\begin{aligned} p(z_u = 0|\mathbf{y}; R, M) &= \sum_{\mathbf{o}_u} p(z_u = 0|s_{1,u} = 1, \ldots, s_{K,u} = 1, \\ &\qquad o_{1,u}, \ldots, o_{K,u}) p(o_{1,u}, \ldots, o_{K,u}; R, M) \\ &= (1 - q_2) \prod_k \tilde{o}_{k,u} + (1 - q_1) \sum_{\mathbf{o}_u \neq \mathbf{1}} \prod_k p(o_{k,u}). \end{aligned} \tag{9}$$

Since, in practice, $q_1 \gg q_2$, in our implementation we only compute the first term, avoiding a summation whose complexity is exponential in the number of occluding layers at u.

Although observation regions for several people may intersect, our tracking algorithm will track these people as distinct targets if their activity zones are not linked by an edge. For example, if two people are proximate in the image plane, but have different depth-order with respect to the same relocatable occluder, their separate identities will be preserved by our tracker.

We conclude this section by briefly noting similarities with prior work. In [12], a person was approximated by a rectangle, and a generative model was developed to produce "ideal random images." A pseudodistance between those images and the actual binary observations was used in constructing the likelihood of a person's location. In [46], [45], an occlusion-sensitive body-part-configuration likelihood was introduced. The image of each body part was divided into three disjoint sets of pixels: those "underneath" the part, Ω_1, those in its immediate vicinity, Ω_2, and the rest, Ω_3. One could also model $\Omega_1 \cup \Omega_2$ by blurring an occlusion mask. The notion of a positive center and inhibitory frame also appeared in [38]. In our case, the union of all observation regions acts as an inhibitory frame since we want to account for an unknown number of people.

3.4 Tracking People Using the Global Scene Model

Given the above layers of graphical models representation, we now turn our attention to tracking a variable number of people around relocatable occluders.

Formulation. Here, we present a deterministic smoothing approximation that operates on a window of T image frames; in our complete system, this formulation is applied in a sliding-window fashion. To accommodate people entering or leaving the scene within the temporal window, we augment our global scene representation with an additional virtual activity zone. The virtual activity zone can accommodate multiple people, and a person's track may enter or exit this zone at any time; since people in the virtual activity zone are not visible, it has no corresponding observation region. We summarize our notation for tracking in Table 3. Let $\mathbf{Y}_t$ be the location of all people at time t, and let $\mathbf{Y}_{1:T}$ be shorthand for $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$. The quantity of interest is the posterior distribution $p(\mathbf{Y}_{1:T}|Z_{1:T})$, which is proportional to the likelihood of the multiperson state multiplied by the prior:

$$p(\mathbf{Y}_{1:T}|Z_{1:T}) \propto p(Z_{1:T}|\mathbf{Y}_{1:T}) p(\mathbf{Y}_{1:T}). \tag{10}$$

We want to approximate the posterior distribution by a point estimate that yields a maximum. However, since trajectories are coupled via an exclusive zone occupancy constraint, maximizing the posterior jointly would be intractable given the number of zones and potential trajectories. As suggested in [12], we estimate trajectories sequentially. Given a person's Markov process on activity zones, the probability of a single trajectory $y_{1:T}$ given $\mathbf{Y}_{1:T}$, the set of already-found trajectories, and $Z_{1:T}$, image evidence, can be written as

$$\begin{aligned} p(y_{1:T}|Z_{1:T}; \mathbf{Y}_{1:T}) &\propto p(Z_{1:T}|y_{1:T}; \mathbf{Y}_{1:T}) p(y_{1:T}) \\ &= p(y_1) p(Z_1|y_1; \mathbf{Y}_1) \prod_{t=2}^{T} \underbrace{p(y_t|y_{t-1}; \mathbf{Y}_t)}_{\text{zone transition}} \underbrace{p(Z_t|y_t; \mathbf{Y}_t)}_{\text{image evidence}}. \end{aligned} \tag{11}$$

Given the recursive dependency between time slices, the most probable trajectory,

$$\hat{y}_{1:T} = \arg\max_{y_{1:T}} p(y_1) p(Z_1|y_1; \mathbf{Y}_1) \prod_{t=2}^{T} p(y_t|y_{t-1}; \mathbf{Y}_t) p(Z_t|y_t; \mathbf{Y}_t), \tag{12}$$

can be found efficiently using the Viterbi algorithm.

As was mentioned in Section 3.3, our approach to account for image evidence handles the case when multiple people occupy distinct activity zones, but their observation regions overlap in the image plane. In many cases, such as when these people are separated by a relocatable occluder, the resulting zone transition graph will ensure they are tracked as distinct targets.

Practical considerations. While in principle it may be possible to directly implement (12), we have found it advantageous to adopt two enhancements. First, as is the common practice, we maximize the log-likelihood of the track $\mathcal{L}(y_{1:T}; \mathbf{Y}_{1:T}, Z_{1:T})$ obtained by taking the logarithm of (12). Second, we extend $\mathcal{L}$ by introducing multiplicative weights for the image-evidence and activity-zone transition terms. These weights allow us to tune the performance of our tracker for the challenging scenario caused by low image resolution and contrast. With this extension, our track log-likelihood becomes

$$\begin{aligned}\mathcal{L}(y_{1:T}; \mathbf{Y}_{1:T}, Z_{1:T}) = \ell_{\text{init}}(y_1) + \ell_{\text{img}}(y_1; \mathbf{Y}_1, Z_1) \\ + \sum_{t=2}^{T} \{\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) \\ + \ell_{\text{img}}(y_t; \mathbf{Y}_t, Z_t)\}.\end{aligned} \tag{13}$$

In (13), we define $\ell_{\text{init}}(y_1) = 0$ if y_1 corresponds to the virtual activity zone and $-c_0$ otherwise. We define $\ell_{\text{img}}(y_t; \mathbf{Y}_t, Z_t) = c_1 \log p(Z_t | y_t, \mathbf{Y}_t)$. To design the transition likelihood, $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t)$, we consider four cases. When y_{t-1} corresponds to the virtual activity zone, but y_t does not, we define $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) = -c_0$ as before. When neither y_t nor y_{t-1} correspond to the virtual activity zone and the activity-zone occupancy constraint is not violated, we define $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) = c_2 \log p(y_t | y_{t-1})$; the limit on activity-zone occupancy is enforced by defining $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) = -\infty$ if y_t is already occupied by $\mathbf{Y}_t$. When a person transitions into the virtual activity zone, we define $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) = -c_3$, and when a person remains in the virtual activity zone we define $\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) = -c_4$. While the resulting set of multiplicative weights may not be the only way to extend $\mathcal{L}$ for our challenging scenarios, it has the advantage of simply enumerating all cases of interest.

We learn $\mathbf{c} = c_0, \ldots, c_4$ from a set of training samples, comprising triplets $y^+_{1:T}, y^-_{1:T}, Z_{1:T}$, where for each triplet we require that $\mathcal{L}(y^+_{1:T}; Z_{1:T}) > \mathcal{L}(y^-_{1:T}; Z_{1:T})$. Finding a feasible $\mathbf{c}$ is then formulated and solved as a linear program. This approach to learning $\mathcal{L}$ would have limited practical use if it had to be applied to each temporal window and would be computationally demanding if $\mathcal{L}$ had to be relearned each time a graphical model layer was instantiated or removed from our global scene representation. While the analysis of generalization guarantees is left for future work, in our experiments, we found that a single trained $\mathcal{L}$ tends to work well across different dynamic scenes, with varying numbers of relocatable occluders.

Given $\mathcal{L}$ in (13), our top-level tracking algorithm is straightforward and is comprised of two stages. In the first stage, the algorithm attempts to extend every track from the previous temporal window, starting with the longest track; in the second stage, the algorithm attempts to find new tracks. Each stage of the algorithm terminates once the relative increase in the log-likelihood becomes less than a threshold; thus the top-level algorithm has one tunable parameter.

3.5 Computational Complexity

As shown in [9], the computational complexity of a direct application of the Viterbi algorithm to an observation sequence of length T generated by an HMM with N states is $O(T \cdot N^2)$. To extend this analysis to our person-tracker, we note that if Ω is the set of all the pixels in all of the observation regions, then the computational complexity of evaluating ℓ_{img} is linear in its cardinality, i.e., $O(|\Omega|)$. By the design of our tracking algorithm, the computational cost of estimating K tracks is linear in K. Therefore, the computational complexity of estimating trajectories of K people over T frames on N activity zones is

$$O\left(K\left[\underbrace{T \cdot N^2}_{\text{transitions}} + \underbrace{T \cdot N \cdot |\Omega|}_{\text{image evidence}}\right]\right). \tag{14}$$

The computational complexity of evaluating the image likelihood for each slice of the dynamic programming may be further reduced by sharing computations between the observation regions. In our implementation, a base image likelihood is evaluated once for each time slice and is then used to compute $\ell_{\text{img}}(y; \cdot)$ in a way that only considers the image evidence localized to r_y. This reduces the overall computational complexity to

$$O\left(K\left[T \cdot N^2 + T \cdot \left(\underbrace{|\Omega|}_{\text{base likelihood}} + N \cdot \underbrace{|r_{\text{average}}|}_{\text{local evidence}}\right)\right]\right), \tag{15}$$

where $|r_{\text{average}}|$ is an average size of the observation region.

In a fully optimized implementation, the computational complexity can be significantly less than specified in (15). For example, to extend a pedestrian track one can exploit the pedestrian's mobility constraints to avoid computing activity-zone transitions for the entire parking lot.

3.6 Dealing with Uncertainty

Potential sources of uncertainty in our system include noisy image measurements and imprecise instantiation of model layers. This measurement noise and imprecision in the scene model can propagate into the estimates of the number of pedestrians and their positions in the scene.

In our formulation, noisy image measurements are handled by aggregating image evidence within the observation regions and across time in our Viterbi-based tracker in (12). Uncertainty in the scene representation is handled by explicitly taking the probability of occlusion at each pixel of an observation region into account in (2). This probability of occlusion is based on "soft" occupancy of each layer's occlusion mask in (1), and therefore allows our person-tracker to explicitly account for the layers' positional uncertainty.

Additional steps can be taken to account for the propagation of noise and errors in our formulation. For instance, measurement noise can also be accounted for in our DBN model by using the sum-product algorithm [24] to compute the marginal distributions over activity zones for every video frame.

4 Experiments

We demonstrate our formulation in the domain of parking lot surveillance. As was mentioned in Section 1, parking lots adjacent to office buildings are often surveyed with one or more fixed cameras pointing at different parts of the lot. The cameras tend to be installed on a building at a shallow depression angle to maximize coverage. This results in severe occlusion of pedestrians and vehicles, especially as the distance to the camera increases. If we regard vehicles as relocatable occluders, the scene can be represented as depth-ordered layers of graphical models. If necessary, this representation can be applied independently to each nonoverlapping view.

4.1 Implementation Details

Database of graphical model layers. We define five relocatable object types—sedan, van, hatchback, station wagon, and minivan—and for each type, we define a coarse 3D polygonal mesh. We define a ring of square nonoverlapping activity zones around a vehicle. For the hatchback, the smallest vehicle type, this yields 16 activity zones, and for the remaining types it yields 18 activity zones. Zone transition probabilities are defined so as to make transitions to immediate neighbors equally likely and to disallow jumps to nonneighboring zones; the same rule applies to the global scene representation as defined in Section 3.2.

We construct a database of models by deterministically sampling vehicle poses in the ground plane. We calibrate the camera using the approach of [31]. For each vehicle type, its orientation is sampled at 16 uniformly spaced angles, and its ground-plane coordinates are sampled in the regions corresponding to high-trafficked areas. These high-traffic areas are defined in more detail when we discuss our data sets, but suffice it to say that in our experiments the number of samples of the ground-plane coordinates ranges between 20 and 25, depending on the size of the parking lot. Given a vehicle's pose in the ground plane, we employ computer graphics rendering to obtain the occlusion mask and depth-ordered observation regions. Since the rendered occlusion masks are quite coarse, we blur them before computing the probability of occlusion in each observation region.

Scene update module. Approaches to tracking vehicles using 3D models or 2D masks have been studied [35], [8], [4], [55] and a patch-based appearance modeling method was proposed in [64]; such a vehicle-tracking module can operate alongside our system. To demonstrate our pedestrian tracker, we implemented a module that instantiates layers when a vehicle enters a legal parking spot. The module operates on a sliding window of frames and relies on binary moving-pixel images and multiframe sparse optical flow. The simple algorithm has bottom-up and a top-down stages. In the bottom-up stage, a foreground blob is compared against the likely vehicle masks at this image location. If the blob passes this test, sparse optical flow is used to compute the distribution over the discrete set of mask orientations in the database. In the top-down stage, a candidate database mask attempts to account for image evidence over a window of frames. If the mask is confirmed, it is used to follow a vehicle until it comes to rest; otherwise, the hypothesis of a moving vehicle is rejected.

In our experiments, the initial scene representation was obtained by manually specifying the image location, depth-order, and type of each layer, then looking up the nearest in the image-plane coordinates models from the database. Because parking lots typically empty out at night, a surveillance system that works around-the-clock could be configured to start with an empty scene representation each morning. If there is sufficient pixel resolution, approaches such as [59], [61] may be employed to segment layer masks in the initial frame.

We also implemented a module to uninstantiate layers from the global scene model. This module relies on the same image evidence used for layer instantiation, but the algorithm is reversed. Specifically, during each temporal window we evaluate two hypotheses for each layer. Under the first hypothesis, image evidence is evaluated conditioned on that layer being stationary. Under the second hypothesis, image evidence is conditioned on that layer moving. The motion trajectory for the second hypothesis is deterministically proposed from optical flow. A likelihood-ratio test determines whether or not the layer is removed from the global scene model. In practice, we have found that our uninstantiation module is discriminative enough to detect when a faraway vehicle "un-parks" while not being distracted by pedestrians walking past vehicles.

As the experiments in this section demonstrate, pedestrian detectors are not a good match for our data sets. Therefore, we have to rely on cues compatible with the available image resolution and contrast, such as moving-region size, to "explain away" image evidence unrelated to pedestrians. We employ a heuristic to suppress false tracks due to moving vehicles: Image evidence in an observation region is considered explained away if this observation region overlaps a foreground blob three times its size. In a system engineered as a turnkey solution, this algorithm could be tuned further. In principle, a probabilistic formulation that jointly reasons about all of the unknowns in the scene is expected to be more effective, but the concern is that the computational complexity of inference would make it unsuitable for practical applications, such as real-time surveillance.

Parameter settings. The parameters of our system are fixed across all experiments as follows:

- Binary occlusion masks in the database are blurred with a Gaussian filter whose half width equals 0.1 times the height of the mask.
- To generate Z, i.e., detect moving pixels, we rely on a background subtraction method based on a mixture of Gaussians. We use an implementation provided by the OpenCV library [6] with default parameters, except for `bg_threshold=0.9`. That value was chosen manually to make the background model rapidly adapt to parking and unparking vehicles on the "training" sequence from the PETS 2001 data set 1, camera view 1.
- To model Z, we set $q_1 = 0.6$ and $q_2 = 0.1$.
- To generate optical flow, we use an implementation of [5] with default parameters.
- The maximum number of pedestrians to track simultaneously is set to 10.

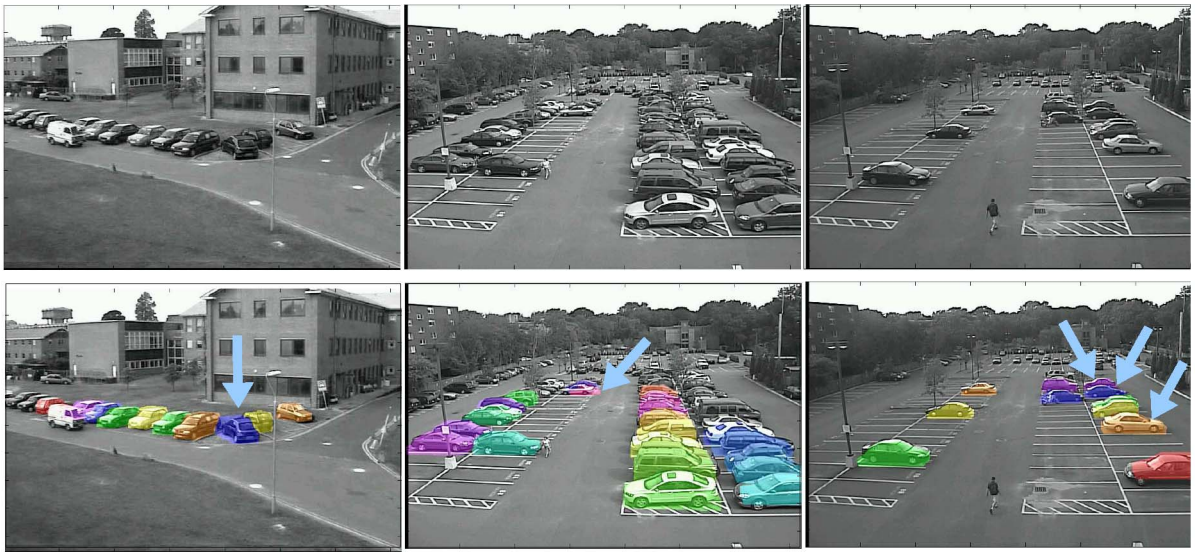

Fig. 8. Top: Representative frames from PETS2001 (left) and COP2007 (center and right) video sequences. Bottom: Vehicle masks from the corresponding global scene models. Arrows in the left two columns highlight automatically instantiated layers; arrows in the right column highlight layers that were automatically uninstantiated later in this video sequence.

TABLE 4
Summary of Test Video Sequences

id	num. frames	source	description
a	2,540	COP2007	two vehicles arrive, drivers and passengers out; two pedestrians get into another vehicle, other pedestrians walk by
b	2,150	COP2007	one vehicle arrives, driver out
c	1,000	COP2007	one vehicle arrives, driver out
d	1,800	COP2007	one vehicle arrives, driver out, other pedestrians walk by
e	2,689	PETS2001	one vehicle arrives, driver out, eight pedestrians walk by
f	11,500	COP2007	Three pedestrians walk to three vehicles and drive off, two pedestrians walk by

- Since the PETS2001 "training" sequence does not have ground-truth bounding boxes, our track likelihood function was trained from 928 samples created with our generative model. For each sample, we first generated a tracklet of length $T = 10$ and its image evidence sequence, then generated an "inferior" tracklet by perturbing the correct one.

4.2 Data Sets[1]

We test our formulation on six video sequences that capture pedestrian and vehicle activities in outdoor parking lots. Typical frames from these sequences and vehicle masks from the corresponding global scene models are shown in Fig. 8. The six video sequences are summarized in Table 4. We next describe our test data in greater detail.

PETS 2001 data set. This data set was originally presented at PETS 2001 [52] and has served as a benchmark for numerous studies, e.g., [44], [43], [67]. We focus on the "testing" sequence from data set 1, camera view 1, since occlusions in that view tend to be more severe. The size of each image frame is 768×576 pixels. The bounding box for the nearest vehicle, which happens to be facing away from the camera, is 66×46 pixels. The bounding box for an unoccluded pedestrian standing next to this vehicle is 15×44 pixels.

1. The data sets and results may be found at http://www.cs.bu.edu/groups/ivc/data/LayeredGraphicalModels.

To generate a database of models for the PETS 2001 sequence, we defined a high-trafficked area to cover the driving lanes and the legal parking spaces. We then deterministically sampled 20 ground-plane locations from this high-trafficked area, and for each location generated one model for each of 16 vehicle orientations and five vehicle types. We stored these models in the database, indexed by the 2D image location, orientation, and vehicle type.

Cambridge Office Park 2007 data set. These sequences were collected at an office park in Cambridge, Massachusetts, during the morning and evening peak hours, and were presented in [14]. The parking lot contains approximately 100 parking spaces. Vehicles parked in the spots toward the front of the parking lot are oriented sideways with respect to the camera. Vehicles parked farther away are either facing the camera or are facing away from the camera. The image size in each of these videos is 720×480 pixels. The projected size of vehicles ranges from 170×70 up front to 54×19 in the middle of the parking lot. An unoccluded person in the middle of the parking lot projects onto a bounding rectangle of size 10×18. In our experiments, we focus on the middle and front portions of the lot.

A single database of models was shared among all of the COP2007 sequences since they all had been captured with the same camera parameters. Because sedans and station wagons in the US tend to be larger than their European counterparts, we enlarged our coarse 3D models for these

Fig. 9. Example frames from our tracking algorithm applied to test video (a), top row, and to test video (e), the PETS 2001 sequence, bottom row. Rectangles indicate observation regions corresponding to the ground-plane zones selected by the tracker; the color of these rectangles in each frame is chosen for visual contrast. A missed pedestrian is marked with a dashed arrow, correct detections are marked with solid arrows.

vehicle types. The high-trafficked area was defined to include the driving lanes and the legal parking spaces in the middle and front portions of the lot. Twenty-five ground-plane locations were deterministically sampled, and then the database of models was generated using the same procedure as for the PETS 2001 data set.

4.3 Qualitative Evaluation

Before conducting the quantitative evaluation of our implementation, we first perform qualitative comparisons with two published methods [44], [51]. The method of [44] employs a deformable-contour model, and qualitative results were published for test sequence (e), the sequence from the PETS 2001 data set. The method of [51] learns flexible sprites and is applied to a portion of test sequence (a), but the results diverge so far from the ground-truth that only a qualitative analysis seems practical.

Comparison with a deformable contour-based tracker. In [44], a B-spline contour was fit to pedestrian-sized foreground blobs and projected onto a learned space of pedestrian outlines. A confirmed pedestrian was tracked frame-to-frame by optimizing her outline from the previous frame to match edge evidence in the current frame; her state was modeled in 3D. Other moving objects in the scene were tracked as regions. Parked vehicles were incorporated into the background, but pixel values occluded by such vehicles were saved; if a parked vehicle moved, the original background was restored.

In [44], this system is applied to sequence (e) from the PETS 2001 data set, but only a qualitative description of the tracker's output at selected frames is provided. This description indicates that the tracker correctly follows isolated pedestrians, e.g., for frames 564 and 975. The driver of a recently parked Peugeot hatchback is tracked from frame 933. The only case of prolonged partial occlusion happens between frames 1,036 and 1,147 when a group of three pedestrians walks between parked vehicles. The description at frame 975 indicates that these three individuals are briefly tracked and the description at frame 1,213 indicates that some of these individuals are tracked, but it is not clear exactly what happens during the period of occlusions. Since [44] does not employ explicit depth-ordering of occluding layers—parked vehicles are merged into the background by design—it may have difficulties in scenarios where partial occlusions are more frequent.

Our scene model is designed to account for image evidence in the vicinity of parked vehicles. As soon as the first pedestrian to enter the scene overlaps one of the observation regions at frame 153, shown in the bottom-left of Fig. 9, she is tracked by our system. At frame 621, our scene update module detects that a recently arrived hatchback is entering a legal parking area. At frame 729, the hatchback is determined to be at rest, and a new graphical model layer is added to our scene representation. Our system tracks the driver of the hatchback starting with frame 933. For the three closely spaced severely occluded pedestrians, three tracks are started at frame 1,089. Although no free-space vehicle-tracking is performed, the instantiated layer's location and orientation is comparable to [35, Fig. 9]; that system uses a 3D model-based ground-plane vehicle tracker with six degrees of freedom.

While it may seem that both the method of [44] and our tracker produce pedestrians' bounding boxes, knowledge of the associated activity zones is helpful. For instance, the driver in sequence (e) is tracked in the activity zones associated with the hatchback and the vehicle to the right of it. This, combined with the knowledge that, in the United Kingdom, a driver sits on the right-hand side, can be used for further semantic analysis, if desired.

Comparison with flexible sprites. In another qualitative study, we compare performance of our method with the well-known sprite-learning approach of [51]. For the purpose of comparison, we selected a 250-frame subsequence from parking lot video (a) where there is substantial pedestrian activity and partial occlusions. The first and last frames of this subsequence are shown in the top-left and

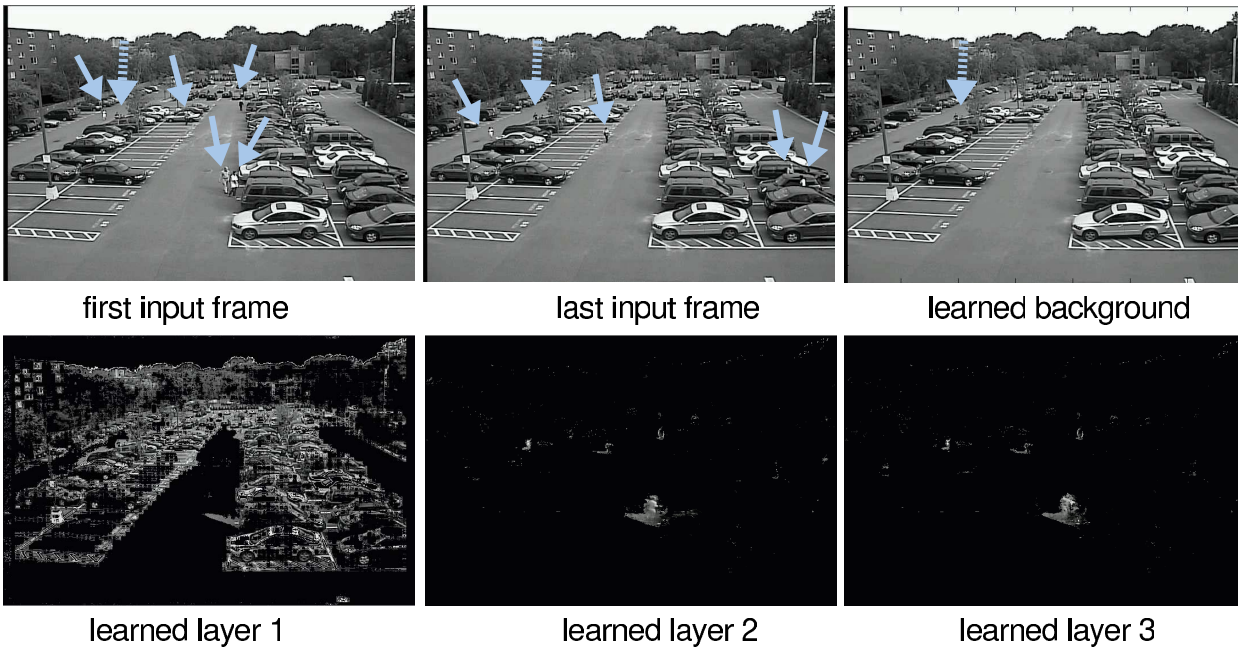

Fig. 10. A flexible sprite learning method is applied to a 250-frame subsequence of test sequence (a) with the first and last frames shown in the top row. Moving pedestrians are highlighted with solid arrows and the stationary pedestrian with a dashed arrow. Although the learned background layer accurately models the stationary background, the three foreground layers do not seem to match individual pedestrians. Please see text for further discussion.

top-center images of Fig. 10, with arrows highlighting the pedestrians' positions.

We use a publicly available implementation of [51] with default parameters, except for the translation window, which is made large enough to track every highlighted pedestrian. The number of foreground layers is limited to three as the computational complexity grows exponentially with the number of layers. Processing our subsequence requires 4.6 hours on an Intel Core2 Quad 2.8 GHz CPU.

In the top-right of Fig. 10, we show the background layer learned by [51]. All pedestrians except for the one highlighted with a dashed arrow, have been correctly removed from the background. Note that the pedestrian considered to be a part of the background is the same one missed by our tracker in Fig. 9.

The bottom row of Fig. 10 shows the three learned foreground layers. The first layer seems to capture abrupt lighting variations in the input video frames, the second layer captures one of the foreground pedestrians and several pedestrians at a distance, and the third layer seems to model the same spatial regions as the second layer. This outcome may indicate that the small apparent size of pedestrians and their prolonged occlusions may not be handled well by a flexible sprite learning method such as [51]. In particular, it may be challenging to employ these results to guide subsequent scene analysis, such as pedestrian-counting or pedestrian-vehicle association. During this subsequence our system tracks five pedestrians: the two occupants of a recently arrived vehicle, who are both occluded from the shoulders down, a person approaching the mid-field from the far end of the parking lot, and two individuals approaching a MINI Cooper in the right-hand portion of the image frame. In the frames preceding this subsequence, two vehicles arrive nearby almost simultaneously. The first one is severely occluded by other vehicles and a tree so its layer is not instantiated. Our method correctly instantiates a layer for the second vehicle and tracks its driver as he exits and then retrieves items from the rear seat.

Comparison with a color and texture-based tracker. In [1], a qualitative comparison with a color and texture-based multitarget tracker was performed using the implementation provided by [49]. It was noted that the lack of color information, low resolution, and severity of occlusions made the COP2007 sequences a poor match for their color/texture-based tracker.

4.4 Quantitative Evaluation

Before presenting a detailed quantitative evaluation of our pedestrian tracker running in parallel with our automatic scene update module, we first evaluate the scene update module.

Evaluation of scene-maintenance module. We evaluate our scene update module with respect to the time of update and the location of the instantiated and uninstantiated occlusion masks. To enable such an evaluation, two human subjects not involved in the algorithm development provided spatiotemporal annotation of the arrivals and departures of vehicles in all of our test video sequences.

We computed absolute differences in video frame indices between our system's estimates and subjective annotations, and computed the F-measure between the bounding boxes of the affected graphical model layers against the bounding boxes marked by the human subjects. The F-measure between the estimated bounding box $\mathcal{E}$ and a ground-truth bounding box $\mathcal{GT}$ was defined in [48] as $\mathrm{F} = \frac{2\,\rho\,\nu}{\rho+\nu}$, where $\rho = \frac{|\mathcal{E}\cap\mathcal{GT}|}{|\mathcal{GT}|}$, $\nu = \frac{|\mathcal{E}\cap\mathcal{GT}|}{|\mathcal{E}|}$, and $|\cdot|$ denotes the area of a bounding box.

As shown in Table 5, the average absolute temporal error of our scene update module is less than two seconds, while the average F-measure is 0.79. Given that the perfect

TABLE 5
Evaluation of the Scene-Maintenance Module

	min	max	average
absolute temporal difference in frames @30fps	4.00	102.00	50.75
F-measure between the bounding boxes	0.57	0.90	0.79

F-measure is 1.0, the performance is good; it also agrees with the examples shown in Fig. 8, where arrows point to the automatically instantiated or uninstantiated layers.

Vehicles entering a parking spot typically decelerate gradually before coming to rest. Our scene update module has to decide when the vehicle has stopped, which is complicated by low resolution and the fact the a vehicle may be "creeping"; hence there seems to be a bias in our system to instantiate models a bit earlier. In fact, without access to vehicle odometry, it was challenging even for our human subjects to pinpoint the precise video frame where a vehicle came to rest. As the F-measure between the ground-truth and the automatically decided bounding boxes indicates, this temporal discrepancy results in a small spatial error.

Evaluation of pedestrian-tracking with changing scene models. Although the PETS2001 data set has served as a benchmark for numerous studies, e.g., [44], [19], [43], [67], there tends to be large variability in evaluation protocols. We adapt the evaluation metrics proposed in [48] and used in [47] that are specifically designed for multi-object tracking systems.

As an overall performance measure, we employ the configuration distance CD^t at time t and the average configuration distance $\overline{\mathrm{CD}}$, computed over a video sequence of length n:

$$\mathrm{CD}^t = \frac{N_{\mathcal{E}}^t - N_{\mathcal{GT}}^t}{\max\left(N_{\mathcal{GT}}^t, 1\right)}, \qquad \overline{\mathrm{CD}} = \frac{1}{n}\sum_{t=1}^{n} |\mathrm{CD}^t|. \tag{16}$$

A perfect tracker yields $\mathrm{CD}^t = 0$ for every t, a missed target at time t results in $\mathrm{CD}^t < 0$, while false tracks or multiple tracks for the same ground-truth target result in $\mathrm{CD}^t > 0$; by construction, $0 \leq \overline{\mathrm{CD}} < \infty$.

We assess the $\overline{\mathrm{CD}}$ for our person-tracking system at different settings of the sliding-window length parameter T for each of the six parking lot video sequences listed in Table 4. The results of this assessment are shown in the graph of Fig. 11: As T increases, the $\overline{\mathrm{CD}}$ tends to monotonically decrease to the global minimum and then it monotonically increases somewhat.

In [12], a single value of $T = 100$ was used, but the reasons for this choice of T were not clear. In our application, the preference toward smaller T may be related to the video camera's frame rate, the average duration of parking and unparking events, and average pedestrian speed.

As a general principle, smoothing with more observations (i.e., increasing T) should tend to improve accuracy; this is typically the case for error measures related to kinematic quantities, e.g., position and velocity. Since CD is a counting error measure, increased T may sometimes work to our advantage and sometimes have the opposite effect. Furthermore, our current implementation of the scene update module computes a tracking estimate for the entire window

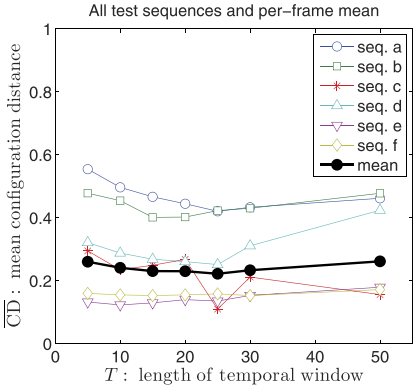

Fig. 11. $\overline{\mathrm{CD}}$ as a function of T on the complete system comprised of a person-tracker and an automatic scene update module. Curves are shown for each of the six parking lot video sequences listed in Table 4. The mean performance curve in the graph is computed using the $\overline{\mathrm{CD}}$ over all frames in the six sequences.

of T frames, and this determines the state space for the person-tracker for these T frames. While in principle the locations of relocatable occluders, the size and the topology of the person-tracking state space, and the locations of an unknown number of persons can be optimized jointly, the resulting inference may be too slow for practical applications such as real-time surveillance; this interesting direction is left for future work.

Comparison with a tracking-by-detection method and with pedestrian detectors. An evaluation with the tracking-by-detection-and-association approach of [60], described in Section 2, was performed; it was done by the authors of [60] themselves. However, it was reported to us that their detection-and-association tracker was not well-suited for our scenarios due to poor resolution, contrast, and, to some extent, strong perspective distortions. The assessment provided by the authors of [60] parallels the observations reported in two other tracking-by-detection approaches. In [62], "... people that are too small in the images (less than 24 pixels in width) are not counted in the evaluation," and in [26] "All images have been processed at their original resolution by SfM and bilinearly interpolated to twice their initial size for object detection."

We next compare our proposed approach with the implicit shape model (ISM) pedestrian detector of [26], described in Section 2, and the Latent-SVM (LSVM) pedestrian detector of [11]. The LSVM approach can be thought of as extending a window-based monolithic detector to a window-based detector informed by a pictorial-structure model of an object; the LSVM detector achieves state-of-the-art results on the PASCAL Visual Object Classes challenge.

For such a comparison, the configuration distance (CD) alone may not provide enough insight into a tracker's performance. Therefore, we adopt a more comprehensive set of performance measures which were originally proposed in [48] and applied to evaluate a tracking system in [47]. These additional performance measures are summarized in Table 6, and are computed for each frame t of the PETS2001 test video sequence. Whether or not a system bounding box matches a ground-truth bounding box is decided by comparing these boxes' F-measure against the

TABLE 6
Summary of the Extended Performance Measures Proposed in [48]

name	valid range	definition
FP^t	$[0, \infty)$	False Positive: a bounding box from a system track does not match any ground-truth bounding boxes.
FN^t	$[0, N_{\mathcal{GT}}^{t,\max}]$	False Negative: a bounding box from a ground-truth track does not match a bounding box of any system track. FN cannot exceed $N_{\mathcal{GT}}^{t,\max}$, the maximum number of ground-truth bounding boxes at time t.
MO^t	$[0, \infty)$	Multiple Objects: a bounding box from a system track matches multiple ground-truth bounding boxes.
MT^t	$[0, \infty)$	Multiple Tracks: bounding boxes from multiple system tracks match a bounding box from a system track.

coverage threshold, $\tau_c \in (0, 1]$. Given these per-frame measures, a system's performance on a test video sequence can be summarized by averaging these measures over all the frames, yielding four nonnegative numbers: $\overline{\mathrm{FP}}$, $\overline{\mathrm{FN}}$, $\overline{\mathrm{MO}}$, and $\overline{\mathrm{MT}}$. One shortcoming of these performance measures is that, in the general case, their average need not equal $\overline{\mathrm{CD}}$.

As mentioned in Section 1, the contribution of our approach is not in free-space pedestrian-tracking. Therefore, to ensure a fair comparison, our person-tracker was not penalized for missing pedestrians whose bounding boxes did not overlap the observation regions of our scene model. Conversely, pedestrian detectors were not penalized for false alarms if the bounding boxes of these false detections did not overlap any of the observation regions of our scene model. In order for the pedestrian detectors [26], [11] to work, each video frame of our test sequences must be upsampled and interpolated by a factor of 2.5.

We apply our evaluation protocol to test sequence (e) from the PETS2001 data set, with a set of coverage test thresholds $\tau_c \in [0.1, 1]$. As Fig. 12 indicates, overall our system based on layers of graphical models performs equally well or better on all four performance measures. The LSVM approach of [11] does not seem to be competitive on this data set. With respect to the ISM approach, our systems achieve uniformly better results with respect to $\overline{\mathrm{FP}}$ as well as $\overline{\mathrm{MO}}$, are competitive in terms of $\overline{\mathrm{MT}}$, and are decisively better in terms of $\overline{\mathrm{FN}}$.

Comparing our system to a system comprised of the same person-tracker but a ground-truth scene update module shows no significant difference. While the system based on automatic scene update has slightly better $\overline{\mathrm{FN}}$ for $\tau_c < 0.5$, the two systems are quite close in terms of performance everywhere else.

By varying τ_c, we change the criterion for a match between an estimated and a ground-truth bounding box. As τ_c increases, the bounding box for a ground-truth track and the bounding box for an estimated track must have a greater overlap to pass the coverage test and be matched. Therefore, when τ_c increases so does the mean false positive, $\overline{\mathrm{FP}}$, as an increased number of estimated tracks do not match the ground-truth tracks. Because our evaluation protocol ignores false negatives (FNs) originating from ground-truth tracks that do not pass the coverage test with at least one observation region of our scene model, $\overline{\mathrm{FN}}$ will tend to decrease as τ_c increases. The multiple objects (MOs) measure decreases because if an estimated track partially overlaps several ground-truth tracks, the overlapping pairs of tracks will fail the coverage test and no MO error will be recorded. Because our tracking algorithm penalizes tracks that attempt to explain the same image evidence, there are few multiple tracker (MT) errors; as τ_c increases $\overline{\mathrm{MT}}$ decreases for the same reasons, as does $\overline{\mathrm{MO}}$.

In summary, the tracking-by-detection approach of [60] was not a good match for this data set, and performance of the LSVM pedestrian detector of [11] was not competitive. Our pedestrian-tracker based on layers of graphical models with automatic scene update performed as well or

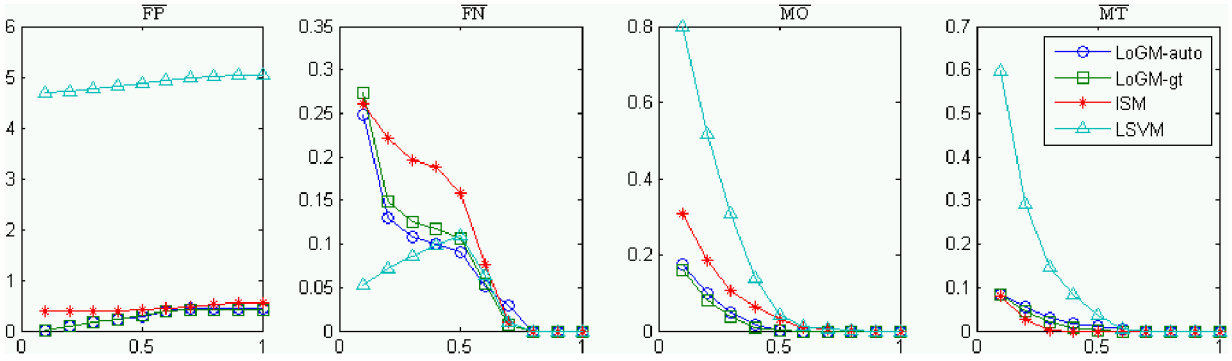

Fig. 12. For test sequence (e), average configuration error measures from [48] are plotted against the coverage-test threshold τ_c. As τ_c increases, the bounding box for a ground-truth track and the bounding box for an estimated track must have a greater overlap to be matched, yielding different error rates. The evaluated approaches are Layers of Graphical Models with our automatic scene update module, Layers of Graphical Models with the ground-truth scene update, ISM pedestrian detector of [26], and Latent-SVM pedestrian detector of [11].

decisively better than the ISM pedestrian detector of [26] on all performance measures.

Throughput. Our video manipulation subsystem runs on .NET and is written in C#. The pedestrian-tracking is implemented in C++ and is called from the .NET platform. Our system runs on a single core of a 2.83 GHz Intel Core2 Quad CPU under Windows 2003 Server OS. Our person-tracker in isolation runs on average at 6.74 Hz on the PETS2001 video sequence cropped to 720×480 pixels, given the foreground moving pixels for every video frame. In theory, background subtraction and other subsystems of our complete system can be run on separate processor cores concurrently with tracking, but implementing this computational model is left for future work.

Our throughput compared favorably with the speed of the competing systems that we evaluated. It was reported in [60] that their tracking-by-detection approach was evaluated on a 3.0 GHz dual-core dual-CPU, and that their implementation utilized all four cores. On their subset of the CAVIAR data set with resolution of 384×288 pixels, they reported an average throughput of 0.27 Hz. The publicly available implementation of the LSVM pedestrian detector was implemented in MATLAB with MEX-calls; it required about 40 seconds to process a video frame of the PETS 2001 test sequence. The publicly available implementation of ISM pedestrian detector was a Linux binary; it required about two minutes per video frame on the PETS2001 test sequence.

5 Conclusions

In our experiments with the parking-lot videos, we have found that the proposed method is able to track pedestrians within the vicinity of parked vehicles despite prolonged, severe occlusions. This level of performance is achieved with the aid of a very simple form of image evidence—raw output of a background subtraction algorithm. Our experiments have demonstrated that in such scenarios, approaches that rely on part-based detectors and on tracking-by-detection do not perform as well as our approach. The experiments have also shown that it is possible to automatically maintain our global scene representation, to change on-the-fly the state space for pedestrian tracking, and to track pedestrians at the same time.

However, our experiments also indicate several areas in need of improvement. The current choice of image features allows our system to cope with the small apparent size of pedestrians, but these features tend to be quite noisy. We believe that this shortcoming may be overcome by optimizing the existing features using training scenarios [58]. Another way of addressing this challenge is to only report a pedestrian's track after she has moved away from a vehicle and is being reliably followed by a free-space pedestrian tracker.

As future work, we aim to develop a formulation for tracking on the boundary of free-space and activity zones, and to validate this formulation with a free-space tracker that is well-matched for our challenging data sets. In addition, our module for instantiating and uninstantiating vehicle models in parking spaces could be primed by a free-space vehicle tracker [35], [34], [4]. One would expect that by integrating observations of a vehicle over time, it may be possible to more accurately predict its type [32] and orientation. A side benefit of employing the free-space pedestrian and vehicle-trackers is the potential to filter out false tracks computed by our system that overlap moving vehicles.

Another promising direction for future research involves extending our formulation to overlapping camera views. A direct extension of our approach is to maintain a separate scene model in each view [21], [53] and fuse the image evidence across all the views [23]. Another approach is to maintain one global 3D representation [54], [23] for relocatable objects and pedestrians. In the 3D case, activity zones around a relocatable object could be defined in the same way as before. A connectivity test to link activity zones of different models could then take into account ground-plane proximity rather than rely on the image-plane cues.

Acknowledgments

This paper reports work that was supported in part by the US National Science Foundation under grants IIS-0713168, IIS-0910908, and IIS-0855065.

References

[1] V. Ablavsky, A. Thangali, and S. Sclaroff, "Layered Graphical Models for Tracking Partially-Occluded Objects," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2008.

[2] M. Andriluka, S. Roth, and B. Schiele, "People-Tracking-by-Detection and People-Detection-by-Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2008.

[3] M. Andriluka, S. Roth, and B. Schiele, "Pictorial Structures Revisited: People Detection and Articulated Pose Estimation," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[4] S. Atev and N. Papanikolopoulos, "Multi-View 3D Vehicle Tracking with a Constrained Filter," *Proc. IEEE Int'l Conf. Robotics and Automation,* 2008.

[5] S.T. Birchfield, "KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker," http://www.ces.clemson.edu/~stb/klt/, 2009.

[6] G. Bradski and A. Kaebler, *Learning OpenCV*. O'Reilly Media, 2008.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2000.

[8] H. Dahlkamp, H.-H. Nagel, A. Ottlik, and P. Reuter, "A Framework for Model-Based Tracking Experiments in Image Sequences," *Int'l J. Computer Vision,* vol. 73, pp. 139-157, 2007.

[9] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* Wiley-Interscience, 2001.

[10] A.M. Elgammal and L.S. Davis, "Probabilistic Framework for Segmenting People under Occlusion," *Proc. Eighth IEEE Int'l Conf. Computer Vision,* 2001.

[11] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, Sept. 2010.

[12] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera People Tracking with a Probabilistic Occupancy Map," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, pp. 267-282, Feb. 2008.

[13] W. Ge and R.T. Collins, "Marked Point Processes for Crowd Counting," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[14] D. Gutchess, V. Ablavsky, A. Thangali, S. Sclaroff, and M. Snorrason, "Video Surveillance of Pedestrians and Vehicles," *Proc. SPIE Conf. Tracking, Pointing, and Laser Systems Technologies XXI,* 2007.

[15] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 809-830, Aug. 2000.

[16] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert, "Recovering Occlusion Boundaries from a Single Image," *Proc. 11th IEEE Int'l Conf. Computer Vision,* 2007.

[17] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 6, pp. 577-589, June 1998.

[18] M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker," *Proc. Eighth IEEE Int'l Conf. Computer Vision,* 2001.

[19] A.D. Jepson, D.J. Fleet, and M.J. Black, "A Layered Motion Representation with Occlusion and Compact Spatial Support," *Proc. European Conf. Computer Vision,* 2002.

[20] N. Jojic and B.J. Frey, "Learning Flexible Sprites in Video Layers," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2001.

[21] J. Kang, I. Cohen, and G. Medioni, "Continuous Tracking within and across Camera Streams," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2003.

[22] S.M. Khan and M. Shah, "A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint," *Proc. European Conf. Computer Vision,* 2006.

[23] S.M. Khan and M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 3, pp. 505-519, Mar. 2009.

[24] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Information Theory,* vol. 47, no. 2, pp. 498-519, Feb. 2001.

[25] M.P. Kumar, P. Torr, and A. Zisserman, "Learning Layered Motion Segmentations of Video," *Int'l J. Computer Vision,* vol. 76, pp. 301-319, 2008.

[26] B. Leibe, K. Schindler, N. Cornelis, and L.V. Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 10, pp. 1683-1698, Oct. 2008.

[27] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *Int'l J. Computer Vision,* vol. 77, pp. 259-289, 2007.

[28] M.J. Leotta and J.L. Mundy, "Predicting High Resolution Image Edges with a Generic, Adaptive, 3-D Vehicle Model," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[29] A. Leykin and R. Hammoud, "Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshop Object Tracking Beyond the Visible Spectrum,* 2006.

[30] Y. Li, L. Gu, and T. Kanade, "A Robust Shape Model for Multi-View Car Alignment," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[31] F. Lv, T. Zhao, and R. Nevatia, "Camera Calibration from Video of a Walking Human," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. 1513-1518, Sept. 2006.

[32] X. Ma and W.E.L. Grimson, "Edge-Based Rich Representation for Vehicle Classification," *Proc. 10th IEEE Int'l Conf. Computer Vision,* 2005.

[33] A. Mittal and L.S. Davis, "M2tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *Int'l J. Computer Vision,* vol. 51, pp. 189-203, 2003.

[34] A. Ottlik and H.-H. Nagel, "Initialization of Model-Based Vehicle Tracking in Video Sequences of Inner-City Intersections," *Int'l J. Computer Vision,* vol. 80, pp. 211-225, 2008.

[35] A. Pece, "Contour Tracking Based on Marginalized Likelihood Ratios," *Image and Vision Computing,* vol. 24, pp. 301-317, 2006.

[36] T. Pollard and J. Mundy, "Change Detection in 3D World," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.

[37] F. Porikli, O. Tuzel, and P. Meer, "Covariance Tracking Using Model Update Based on Lie Algebra," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2006.

[38] C. Rasmussen and G.D. Hager, "Probabilistic Data Association Methods for Tracking Complex Visual Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 560-576, June 2001.

[39] J. Renno, J. Orwell, and G. Jones, "Learning Surveillance Tracking Models for the Self-Calibrated Ground Plane," *Proc. British Machine Vision Conf.,* 2002.

[40] J. Renno, D. Greenhill, J. Orwell, and G. Jones, "Occlusion Analysis: Learning and Utilising Depth Maps in Object Tracking," *Image and Vision Computing,* vol. 26, pp. 430-441, 2008.

[41] M.S. Ryoo and J.K. Aggarwal, "Observe-and-Explain: A New Approach for Multiple Hypotheses Tracking of Humans and Objects," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2008.

[42] S.M. Seitz and C.R. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 1997.

[43] A. Senior, A. Hampapur, Y.-L. Tan, L. Brown, S. Pankanti, and R. Bolle, "Appearance Models for Occlusion Handling," *Image and Vision Computing,* vol. 24, pp. 1233-1243, 2006.

[44] N. Siebel and S.J. Maybank, "Real-Time Tracking of Pedestrians and Vehicles," *Proc. Second IEEE Int'l Workshop Performance Evaluation of Tracking and Surveillance,* 2001.

[45] L. Sigal, "Continuous-State Graphical Models for Object Localization, Pose Estimation and Tracking," PhD dissertation, Brown Univ., 2008.

[46] L. Sigal and M. Black, "Measure Locally, Reason Globally: Occlusion-Sensitive Articulated Pose Estimation," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2006.

[47] K. Smith, S.O. Ba, J.-M. Odobez, and D. Gatica-Perez, "Tracking the Visual Focus of Attention for a Varying Number of Wandering People," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 7, pp. 1212-1229, July 2008.

[48] K. Smith, D. Gatica-Perez, J.-M. Odobez, and S. Ba, "Evaluating Multi-Object Tracking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition Workshop Empirical, Evaluation Methods in Computer Vision,* 2005.

[49] V. Takala and M. Pietikainen, "Multi-Object Tracking Using Color, Texture and Motion," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.

[50] H. Tao, H. Sawhney, and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 75-89, Jan. 2002.

[51] M. Titsias, "Unsupervised Learning of Multiple Objects in Images," PhD dissertation, School of Informatics, Univ. of Edinburgh, 2005.

[52] Performance Evaluation for Tracking and Surveillance (PETS) 2001 Dataset, The Univ. of Reading, UK Std., http://www.cvg.cs.rdg.ac.uk/PETS2001/pets2001-dataset.html, 2001.

[53] L. Vacchetti, V. Lepetit, and P. Fua, "Stable Real-Time 3D Tracking Using Online and Offline Information," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 10, pp. 1385-1391, Oct. 2004.

[54] S. Vedula, P. Rander, H. Saito, and T. Kanade, "Modeling, Combining, and Rendering Dynamic Real-World Events from Image Sequences," *Proc. Fourth Int'l Conf. Virtual Systems and Multimedia,* 1998.

[55] V. Venkataraman, X. Fan, and G. Fan, "Integrated Target Tracking and Recognition Using Joint Appearance," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshop Object Tracking and Classification Beyond the Visible Spectrum,* 2008.

[56] R. Vezzani and R. Cucchiara, "Ad-Hoc: Appearance Driven Human Tracking with Occlusion Handling," *Proc. British Machine Vision Conf. Workshop Tracking Humans for the Evaluation of their Motion in Image Sequences,* 2008.

[57] J. Wang and E. Adelson, "Representing Moving Images with Layers," *IEEE Trans. Image Processing,* vol. 3, no. 5, pp. 625-638, Sept. 1994.

[58] B. White and M. Shah, "Automatically Tuning Background Subtraction Parameters Using Particle Swarm Optimization," *Proc. IEEE Int'l Conf. Multimedia and Expo,* 2007.

[59] J. Winn and J. Shotton, "The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2006.

[60] B. Wu and R. Nevatia, "Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses," *Int'l J. Computer Vision,* vol. 82, pp. 185-204, 2009.

[61] B. Wu and R. Nevatia, "Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature Based Classifier," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.

[62] J. Xing, H. Ai, and S. Lao, "Multi-Object Tracking through Occlusions by Local Tracklets Filtering and Global Tracklets Association with Detection Responses," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[63] M. Xu and T. Ellis, "Partial Observation vs. Blind Tracking through Occlusion," *Proc. British Machine Vision Conf.,* 2002.
[64] Z. Yin and R. Collins, "On-the-Fly Object Modeling While Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.
[65] T. Yu, Y. Wu, N.O. Krahnstoever, and P.H. Tu, "Distributed Data Association and Filtering for Multiple Target Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2008.
[66] Y. Zhou and H. Tao, "A Background Layer Model for Object Tracking through Occlusion," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* 2003.
[67] L. Zhu, J. Zhou, and J. Song, "Tracking Multiple Objects Through Occlusion with Online Sampling and Position Estimation," *Pattern Recognition,* vol. 41, pp. 2447-2460, 2008.

Vitaly Ablavsky received the BA degree in mathematics from Brandeis University in 1992, the MS degree in computer science from the University of Massachusetts Amherst in 1996, and the PhD in computer science from Boston University in 2011. He has taken a postdoctoral fellowship at EPFL's CVLAB. Previously, he was a software/research engineer at Amerinex Applied Imaging, Inc., Cognex Corporation, and Charles River Analytics, Inc. His interests are object tracking and machine learning. He is a member of the IEEE and the IEEE Computer Society.

Stan Sclaroff received the PhD degree from the Massachusetts Institute of Technology in 1995. He is a professor of computer science and the chair of the Department of Computer Science at Boston University. He founded the Image and Video Computing research group at Boston University in 1995. In 1996, he received a US Office of Naval Research (ONR) Young Investigator award and the US National Science Foundation (NSF) Faculty Early Career Development award. Since then, he has coauthored numerous scholarly publications in the areas of tracking, video-based analysis of human motion and gesture, deformable shape matching, and recognition, as well as image/video database indexing, retrieval, and data mining methods. He has served on the technical program committees of more than 90 computer vision conferences and workshops. He has served as an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000-2004 and 2006-present. He is a senior member of the IEEE and a member of the IEEE Computer Society.