

2011-03-16

# Layered graphical models for tracking partially-occluded moving objects in video (PhD thesis)

---

Ablavsky, Vitaly. "Layered Graphical Models for Tracking Partially-Occluded Moving Objects in Video (PhD Thesis)", Technical Report BUCS-TR-2011-010, Computer Science Department, Boston University, March 16, 2011. [Available from: <http://hdl.handle.net/2144/11367>]

<https://hdl.handle.net/2144/11367>

*"Downloaded from OpenBU. Boston University's institutional repository."*



**LAYERED GRAPHICAL MODELS  
FOR TRACKING  
PARTIALLY-OCCLUDED MOVING OBJECTS  
IN VIDEO**

*VITALY ABLAVSKY*

Dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

**BOSTON  
UNIVERSITY**

BOSTON UNIVERSITY  
GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

**LAYERED GRAPHICAL MODELS  
FOR TRACKING  
PARTIALLY-OCCLUDED MOVING OBJECTS  
IN VIDEO**

by

**VITALY ABLAVSKY**

B.A., Brandeis University, USA, 1992  
M.S., University of Massachusetts Amherst, USA, 1996

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

2011

© Copyright by  
VITALY ABLAVSKY  
2011

Approved by

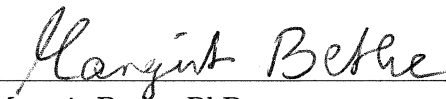
First Reader



---

Stan Sclaroff, PhD  
Professor of Computer Science  
Boston University

Second Reader



---

Margrit Betke, PhD  
Associate Professor of Computer Science  
Boston University

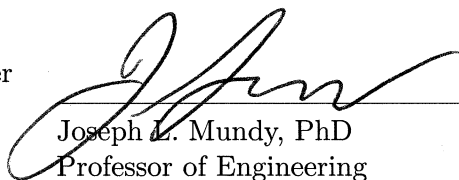
Third Reader



---

Erik Learned-Miller, PhD  
Associate Professor of Computer Science  
University of Massachusetts Amherst

Fourth Reader



---

Joseph L. Mundy, PhD  
Professor of Engineering  
Brown University

## Acknowledgments

I would like to thank Stan Sclaroff for advising me during my entire PhD journey. Stan’s time commitment to his students, his guidance in developing useful ideas, and his mentorship in lucidly conveying these ideas, have made my apprenticeship a uniquely enriching experience. I am also grateful to Stan for imbuing the Image and Video Computing group with a hard-working yet collaborative spirit.

I thank the faculty members who provided feedback on my thesis at several key stages. In particular, I thank Margrit Betke for her constructive critique during the prospectus defense and for a careful reading of my complete thesis. I thank Erik Learned-Miller for challenging me to think broadly about key aspects of my formulation, and for suggesting practical directions for future research. In response to Joseph Mundy’s gentle but probing questions, there is now a clarified and expanded discussion of the complete system.

I thank my comrades: Ashwin—for infinitely many discussions on as many topics; Quan—for the conversations about parameter-sensitive detectors and about life in general; and Liliana—for her enthusiasm and for reading the first draft of my first journal paper. I would like to thank everyone who overlapped with me in space-time at IVC for enriching my experience here. The list includes, in alphabetical order, Alex S., Angshuman, Bill, Chris, Diane, Eric, Esra, Gökberk, Gordon, Hee Deok, Jared, Javier (Flavio), John, Joni, Leyong (Alex), Maria, Murat, Nazli, Rómer, Rufat, Rui, Sam, Tai-Peng, Tianqiang, Vasilis, Wajeaha, Walter, and Zheng, as well as this year’s newcomers, Danna, Kun, Mike, Qinxun, and Shugao. Outside IVC, *mille grazie a Flavio!*

I dedicate this thesis to Abby, the love of my life. She is, to borrow a term from Italo Calvino, my *Lettrice* and a traveller through the physical and the imaginary worlds. Her dedication and personal sacrifices while I pursued my PhD work could not possibly be summarized in this short note of infinite gratitude.

I thank my family for doing their best, raising me in a country whose name has since been relegated to archives. I am grateful to them for finding a way to escape from the void, at tremendous personal cost, and with the hope of normal lives for their children.



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 low-resolution, partially-occluded pedestrians in the vicinity of parked vehicles. In these scenarios some tracking formulations that rely on part-based person detectors may fail completely. Our pedestrian tracker fares well and compares favorably with the state-of-the-art pedestrian detectors—lowering false positives by twenty-nine percent and false negatives by forty-two percent—and a deformable-contour-based tracker.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Main Contributions . . . . .	2
1.3	Plan of the Thesis . . . . .	4
1.4	Publications . . . . .	6
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	Representations of Dynamic Scenes . . . . .	7
2.2	Tracking of People . . . . .	11
2.3	Recognition of Actions . . . . .	14
2.4	Advantages to the Proposed Approach . . . . .	16
<b>3</b>	<b>Scene Representation: Layered Graphical Models</b>	<b>19</b>
3.1	Basic Idea . . . . .	19
3.2	Local Representation: Graphical-Model Layer . . . . .	21
3.3	Global Scene Model: Depth-Ordered Layers of Graphical Models . . . . .	23
3.4	Accounting for Image Evidence in the Depth-Ordered Layers of Graphical Models . . . . .	26
<b>4</b>	<b>Tracking a Variable Number of People with Layered Graphical Models</b>	<b>31</b>
4.1	Tracking People with Reversible Jump Markov Chain Monte Carlo . . . . .	32
4.2	Tracking People with the Viterbi Algorithm . . . . .	36
4.3	Comparison of the RJ MCMC-based and the Viterbi-based Tracking Algorithms . . . . .	41

<b>5 Experiments with the RJ MCMC-based Tracker</b>	<b>43</b>
5.1 Datasets and Implementation Details . . . . .	43
5.2 Qualitative Evaluation . . . . .	45
<b>6 Experiments with the Viterbi-based Tracker</b>	<b>52</b>
6.1 Scene Update Module . . . . .	52
6.2 Implementation Details . . . . .	57
6.3 Datasets . . . . .	58
6.4 Qualitative Evaluation . . . . .	60
6.5 Quantitative Evaluation . . . . .	64
6.6 The Effect of Model Uncertainty on the Pedestrian Tracker . . . . .	72
<b>7 A Guide to Applying Layered Graphical Models to New Problem Domains</b>	<b>86</b>
7.1 Designing Graphical-Model Layers . . . . .	87
7.2 Designing a Scene-Maintenance Algorithm . . . . .	92
7.3 Designing a Person Tracker . . . . .	94
<b>8 Discussion and Conclusions</b>	<b>98</b>
8.1 Main Contributions . . . . .	98
8.2 Future Work . . . . .	98
<b>References</b>	<b>101</b>
<b>Curriculum Vitae</b>	<b>110</b>

## List of Tables

2.1	Comparison of scene representations. . . . .	17
3.1	Notation for graphical-model layer formulation. . . . .	21
3.2	Notation for accounting for image evidence. . . . .	27
4.1	Notation for RJ MCMC-based tracking formulation. . . . .	33
4.2	Notation for Viterbi-based tracking formulation. . . . .	37
6.1	Summary of test video sequences. . . . .	60
6.2	Evaluation of the scene-maintenance module. . . . .	65
6.3	Summary of the extended performance measures. . . . .	68

# List of Figures

1-1	Examples of occluders. . . . .	1
3-1	Basic idea of the approach. . . . .	20
3-2	System diagram for a tracking application. . . . .	20
3-3	Graphical-model layer. . . . .	22
3-4	A Dynamic Bayesian Network view of a graphical-model layer. . . . .	24
3-5	Depth-ordered graphical-model layers. . . . .	24
3-6	An example of instantiated graphical-model layers. . . . .	26
5-1	Selected video frames and observation regions from the Computer Laboratory dataset. . . . .	44
5-2	Results of RJ MCMC on the Computer Laboratory sequence. . . . .	45
5-3	Evaluation of RJ MCMC on the Computer Laboratory sequence. . . . .	47
5-4	Results of RJ MCMC on the MINI Cooper sequence. . . . .	49
5-5	Results of RJ MCMC on the “far-field” sequence. . . . .	50
6-1	Representative video frames from test video sequences. . . . .	59
6-2	Example frames from our tracking algorithm applied to test video. . . . .	62
6-3	Example frames from a flexible-sprite-learning method applied to test video. . . . .	63
6-4	Configuration distance error measure as a function of the number of video frames in our sliding-window tracking algorithm. . . . .	66
6-5	Comparison of our approach with pedestrian detectors using extended performance measures. . . . .	69
6-6	Effects of location uncertainty . . . . .	77
6-7	Distribution of $\Delta\theta$ . . . . .	79

6-8	Effects of orientation uncertainty. . . . .	81
6-9	Effects of occluder-class uncertainty. . . . .	85

## List of Abbreviations

2D	.....	Two-Dimensional
3D	.....	Three-Dimensional
CD	.....	Configuration Distance
DBN	.....	Dynamic Bayesian Network
EM	.....	Expectation-Maximization
FP	.....	False Positive error
FN	.....	False Negative error
HMM	.....	Hidden Markov Model
MO	.....	Multiple Objects error
MT	.....	Multiple Trackers error
RJ MCMC	.....	Reversible Jump Markov Chain Monte Carlo
SNR	.....	Signal-to-Noise Ratio

## Chapter 1

# Introduction

### 1.1 Motivation

Tracking multiple targets using fixed cameras with non-overlapping 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 [Wu and Nevatia, 2009, Li et al., 2009, Andriluka et al., 2009] and toward developing tracking algorithms that can cope with a short-term loss of observations [Betke et al., 2007, Zhu et al., 2008, Ryoo and Aggarwal, 2008, Xing et al., 2009, Papadourakis and Argyros, 2010]. A complementary line of research has focused on learning static occlusion maps using large sets of observations accumulated over time [Renno et al., 2007]. In this thesis 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



**Figure 1-1:** 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.

*relocatable objects or relocatable occluders.*

Scenarios that include relocatable occluders are quite common. Fig. 1-1 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 non-overlapping 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, 3D model-based trackers of [Pece, 2006, Dahlkamp et al., 2007] are likely to be distracted by inter-vehicle occlusions. Furthermore, the image resolution typical in such scenarios makes it difficult to apply 3D alignment techniques based on high-contrast edges [Leotta and Mundy, 2009].

## 1.2 Main Contributions

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 on-line 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 [Li

et al., 2009].

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

In our representation, a scene is modelled 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 comprises 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. Lastly, 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* [Ablavsky et al., 2008] 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 thesis is *not* about. This thesis 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 thesis is not about learning a static occlusion map. Methods for learning such maps [Renno et al., 2007, Xu and Ellis, 2002, Hoiem et al., 2007] are complementary to our approach. This thesis is not about free-space tracking, for which off-the-shelf algorithms of [Takala and Pietikainen, 2007, Smith et al., 2008, Fleuret et al., 2008] can be applied. This thesis 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.

Lastly, we assume that the only source of information is a single fixed camera or a set of fixed cameras with non-overlapping 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 multi-view scenarios is a promising direction for future research, but it is not within the scope of this thesis.

### 1.3 Plan of the Thesis

The remainder of the thesis is as follows:

Chapter 2 reviews the state of the art in scene-modelling for tracking people, algorithms for tracking multiple persons, and approaches to action recognition that are relevant to

our proposed scene representation. We highlight the advantages and disadvantages of the image-plane, ground-plane, and 3D representations. The chapter concludes with the discussion of the benefits for our *implicit*-3D scene representation.

Chapter 3 defines our proposed scene model—layers of graphical models. We first define a graphical-model layer that represents a person’s motion-patterns and appearance with respect to a relocatable occluder; we also introduce a database of graphical-model layers. Next, we define our scene representation comprising graphical-model layers retrieved from the database, instantiated in image coordinates, and depth-ordered. In the remainder of this chapter we derive a generative model to account for image evidence; this model is applicable to tracking scenarios with low resolution and poor contrast.

Chapter 4 derives two algorithms for tracking a variable number of people in the vicinity of the instantiated graphical-model layers. The first algorithm is stochastic and is based on Reversible Jump Markov Chain Monte Carlo (RJ MCMC). The second algorithm is deterministic and is based on computing the optimal paths in the trellis defined by our scene representation; the optimization is accomplished via the Viterbi algorithm. For each tracking algorithm we derive its computational complexity. The chapter concludes with our assessment of the relative advantages of each of the two tracking approaches.

Chapter 5 demonstrates our RJ MCMC-based tracking algorithm in the indoor and outdoor scenarios. In the indoor scenario people move inside a computer laboratory and are partially occluded by the desks and workstations. In the outdoor scenario pedestrians move in the parking lot and are partially occluded by the parked vehicles. The chapter concludes with a qualitative evaluation of the tracking results and the recommendation that the RJ MCMC-based algorithm might not be the preferred choice for scenarios with low image resolution and poor contrast.

Chapter 6 demonstrates our Viterbi-based tracking algorithm in a challenging scenario—parking-lot surveillance. We first qualitatively compare the performance of our system against a color-and-texture based tracker, a contour-based tracker, and a layer-learning algorithm based on Expectation-Maximization. Next we quantitatively compare our ap-

proach against tracking-by-detection using part-based models and with two scanning-window pedestrian detectors. The remainder of chapter 6 analyzes the uncertainty in our scene model and the propagation of uncertainty into the estimates of the number and the location of pedestrian computed by our Viterbi-based tracking algorithm.

Chapter 7 is a guide to applying our scene representation to new problem domains. First, we provide a recipe for modelling domain-specific relocatable occluders as graphical-model layers. Second, we discuss the issues of scene maintenance—dynamically instantiating and removing graphical-model layers. Third, we define the criteria for selecting a person-tracking algorithm. For each of the three aspects of our scene representation we provide recommendations for specifying and learning the corresponding model parameters.

Chapter 8 is a discussion of the advantages of the proposed approach and recommendations for future work. These recommendations include ideas for fusing multiple trackers and ideas for improving the scene-maintenance module.

## 1.4 Publications

This thesis is based in part on the following publications:

- Vitaly Ablavsky, Ashwin Thangali, and Stan Sclaroff. Layered graphical models for tracking partially-occluded objects. In CVPR, 2008 [Ablavsky et al., 2008].
- Vitaly Ablavsky and Stan Sclaroff. Layered graphical models for tracking partially-occluded objects. T-PAMI, 2011 [Ablavsky and Sclaroff, 2011].

## Chapter 2

### Related Work

In this chapter we review related *representations* for dynamic scene analysis, then discuss related *approaches* to tracking persons and vehicles, and highlight relevant approaches to recognizing actions of people with scene context.

#### 2.1 Representations of Dynamic Scenes

We begin our review of image-plane representations with the  $W^4$  system [Haritaoglu et al., 2000] as its scope extended beyond tracking and its tracking accuracy compared favorably to the state of the art. Indeed, the  $W^4$  system demonstrated tracking and activity analysis of pedestrians walking in isolation or in groups; the input to  $W^4$  was a video sequence captured by a single grayscale camera. To accomplish this functionality,  $W^4$  comprised multiple sub-systems, including foreground-region clustering, body-part detection of isolated pedestrians, head detection of pedestrians in a group, and symmetry analysis for carried-object detection. 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. Although pedestrian sizes as low as 25 x 10 pixels were reported in [Haritaoglu et al., 2000], it is not clear how the boundary analysis performed in cases of reduced resolution or low contrast; indeed, it was acknowledged that the resolution of 75 x 50 pixels was preferred. Because the  $W^4$  system did not infer depth-order of overlapping pedestrians it may have had difficulties tracking through prolonged occlusions or with recognizing multi-person interactions characterized by changes in their depth-order. Because experimental validation focused on occlusions of a person by other people, it is also not clear how the system would handle

static or relocatable occluders in the scene; indeed the paper tends to give few details regarding challenges posed by vehicles in parking-lot scenes.

To extract and depth-order multiple moving regions from an uncalibrated video sequence, a layered representation has been proposed [Wang and Adelson, 1994]. In this representation an object in a scene was modelled as a 2D textured region and an *alpha* map indicating transparency at each pixel. Given such a representation, a single image frame was generated by applying composition rules to the depth-ordered textured regions and their alpha maps. To extend the representation to sequences of images, each layer was associated with a 2D velocity map. By applying the composition algorithm to depth-ordered layers undergoing the motions specified by their velocity maps, a video sequences could be generated. The inverse problem of inferring parameters of a layered representation from a single video sequence was proposed, one motivating application being video compression. To solve this inverse problem, 2D apparent velocities of pixels were computed using the brightness-constancy constraint. These velocities were assumed to be generated by multiple layers, whose number was specified as an input to the algorithm. A robust motion segmentation algorithm in the affine space of motion parameters was combined with k-means clustering and followed by hypothesis testing to determine each pixel’s membership with respect to its layer.

It was observed in [Irani and Anandan, 1998] that recovering a layered representation with 2D motion models might not always be practical. In particular if a camera undergoes an arbitrary motion while capturing a multi-planar scene comprising numerous small objects at different depths and a static background, approaches based on stabilizing with respect to a dominant motion are likely to fail. Therefore, a stratified approach to decompose an image into coherent moving regions was proposed. This approach started with a global 2D motion model, and if this model did not explain all the image evidence, the approach progressed to the 2D layered model; if that 2D layered model did not explain all of the remaining image evidence, the approach progressed toward a 3D-parallax model. In experiments this formulation was applied to moving-object detection in aerial and street-level

video sequences.

To allow layers to undergo deformations between image frames, [Jojic and Frey, 2001] proposed a *flexible sprite* scene representation. Following [Wang and Adelson, 1994] a layer was defined as a textured region and a transparency mask. A scene at each pixel was defined as a multiplicative superposition of the front-most layer and the background, which itself could be decomposed further into layers. To allow the texture of a layer to undergo arbitrary appearance variations and for its mask to vary in shape, both the texture region and the transparency mask were modelled as multi-dimensional Gaussian random variables; the joint distribution over the texture and the occlusion mask was written as a product of these two Gaussian distributions. A class label was introduced, modelled as a discrete random variable, to enable class-conditional modelling of each flexible sprite. In experiments the the number of classes was given to the algorithm as an input parameter; in typical test video sequences the number of sprites was three. Inference over all the parameters—sprite’s texture, its occlusion mask, and their spatial translations at each image frame—was performed using variational Expectation-Maximization; a frame rate of 1Hz was reported on 320 x 240 image sequences with three layers. However, it was not clear from the experimental validation how the algorithm would perform on video sequences comprising a large number of objects of different sizes and with varying poses.

The family of spatial transformation that a layer could undergo was limited to translations in [Jojic and Frey, 2001]. The family of allowed spatial transformations was extended to the affine family in [Winn and Blake, 2004]. Layered representations with stronger segmentation priors were proposed in [Jepson et al., 2002, Tao et al., 2002, Kumar et al., 2008]. With the exception of [Tao et al., 2002], 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 [Titsias, 2005], but the method might still be impractical for a real-time system.

To enable modelling of more realistic interactions between the moving objects in the

scene and a static background, [Zhou and Tao, 2003] proposed to model the background not as a single layer but as a collection of layers. To accomplish this, the foreground and the background layers were interleaved. The front-most layer was designated as the first occluding-background layer; it was followed by the first foreground layer, which was then followed by the second background layer, etc. In the complete system that inferred the number of layers, each newly created foreground layer was associated with a newly created background layer; both were added to the scene representation in the appropriate depth-order. All background layers shared the same motion model; each foreground was associated with its own motion model. In outdoor experiments this scene representation was effective for tracking a vehicle as it drove behind a tree and was partially occluded by undulating terrain.

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. While the scene representation of [Zhou and Tao, 2003] was applicable to a larger set of real-world scenarios than some of the prior works on layers, it did not address the motion-around-layers aspect of scene modelling.

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 [Renno et al., 2002] 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 [Renno et al., 2007] this method was extended to cope with static occluders, such as 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 comprising relocatable occluders. An approach to handle static occluders was presented in [Vezzani et al., 2010] 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 comprises objects of known types, e.g., vehicles, it may be practical to acquire their 3D models off-line. The scene can then be represented by instances of 3D models whose pose varies over time. Such model-based tracking has been applied to vehicles [Pece, 2006, Dahlkamp et al., 2007] and pedestrians [Leykin and Hammoud, 2006, Leibe et al., 2008]. 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 a target’s position estimation in every frame, a volumetric representation may be appropriate. In [Seitz and Dyer, 1997, Vedula et al., 1998] voxel-carving of a bounded 3D volume seen from multiple calibrated views was demonstrated, while in [Pollard and Mundy, 2007] 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 [Haritaoglu et al., 2000, Smith et al., 2005b] or an ellipse [Comaniciu et al., 2000] if tracking in the image coordinates, or a 3D bounding box [Fleuret et al., 2008] 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 [Haritaoglu et al., 2000], a color histogram in [Fleuret et al., 2008], or region covariance in [Porikli et al., 2006]. However, in some multi-view approaches [Khan and Shah, 2006, Khan and Shah, 2009] 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 person's model into sub-regions 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 [Isard and MacCormick, 2001]. A person was modelled as a generalized cylinder, and her color appearance in the image plane was modelled as a grid of uniformly-spaced disks. In [Kang et al., 2003] each foreground blob was partitioned into regions in a polar coordinate system and the color distribution of each region then estimated. In the multi-view approach of [Mittal and Davis, 2003] a person was modelled 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 non-trivial to design a partitioning that anticipates all possible variations in a target's appearance.

In order to better cope with inter-person occlusions or to meet requirements of a specific application, methods that align a part-based model to each tracked person have been proposed. In [Smith et al., 2008], which tracked pedestrians passing by a store-window display to determine their focus of visual attention, the target model comprised 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 [Elgammal and Davis, 2001], where a person was modelled in the image plane as an ellipse that was partitioned based on image evidence into horizontal slices corresponding to head, torso, and leg regions. In [Leibe et al., 2007, Leibe et al., 2008] local appearance was modelled via a codebook, and a pedestrian's shape defined implicitly via the spatial probability distribution over the codebook's entries.

Methods that combine discriminatively-trained whole-body or body-part detectors in a tracking framework have compared favorably with the state of the art. In detection-based tracking approaches, the ability to accurately model the background and detect

moving pixels tends to be less important; indeed many such tracking approaches do not use background subtraction. Therefore, detection-based trackers may be applicable when the motion of the camera and the amount of change in a scene make background subtraction impractical. On the other hand, person detectors trained for one scenario might not always generalize to a novel viewpoint, scenario, or an application domain, and might require pixel resolution that is not supported by the available imaging sensor.

In [Yu et al., 2008] a multi-tracking algorithm was proposed where data association and target-state estimation were combined in a variational Expectation-Maximization formulation. In this formulation, a target’s state was regarded as a continuous missing variable, optimized during the E-step, while the association was performed during the M-step using a graph-based algorithm, such as belief propagation. This tracking formulation was then applied to tracking people in video sequences from a single calibrated camera by relying on body-part detectors as image evidence. These body-part detectors were trained for a person’s head-and-shoulders, the torso, and the legs; some false alarms were discarded based on known 3D scene geometry. A depth-based body-part association prior was defined, so that, for example, the head of a person closer to the camera would not be associated with a person hypothesized behind her.

Multi-person tracking with an articulated human-body model was proposed in [Andriluka et al., 2008]. A person detector is derived in such a way as to identify a limb-based structure inside the detection window. To detect and track people in short sequences, the dynamics of a person’s limbs was modelled via a low-dimensional representation obtained from a hierarchical Gaussian-process latent variable model. These short-term *tracklets* were assembled into longer tracks, one at a time, via the Viterbi algorithm. State-of-the-art results were demonstrated on several TU Darmstadt datasets, including the Campus video sequence, where the resolution of pedestrians ranged from 37 x 145 to 80 x 310 pixels.

State-of-the-art tracking results were demonstrated in a tracking-by-detection approach of [Wu and Nevatia, 2009] which relied on body-part detections obtained via Adaboost. In [Xing et al., 2009] multi-view and multi-part person detectors were used in a global

optimization framework.

An approach to deal with multiple sources of uncertainty in a detector-based tracker was proposed in [Breitenstein et al., 2011]. In addition to relying on a person detector’s binary output, the proposed formulation utilized the detector’s confidence in its decision. These detector confidences were treated as additional evidence by the *graded* observation model; in addition to generic object detectors, on-line instance-specific models were trained. Tracking was posed as a filtering problem so that only information from the past needed to be considered.

### 2.3 Recognition of Actions

In this thesis we demonstrate the effectiveness of our scene representation for tracking in the parking-lot scenarios. However, in many applications tracking is followed by action- or activity-recognition. In this section we highlight the relevant approaches where knowledge of the scene is exploited for recognition of human activity.

Given motion patterns of people interacting with a static environment, it may be possible to automatically label regions in this environment with typical human activities. In [Demirdjian et al., 2002] for each tracked person in an office environment their ground-plane velocity and height above the ground plane were sampled in time. These samples were clustered with respect to spatial attributes to yield a set of activity zones. A proposed application of such activity zones was demonstrated for an office environment where ambient parameters, e.g., lighting, were automatically controlled based on human activities. In [Peursum et al., 2005] regions of an office environment were labelled using the articulated pose of a human subject; robustness to occlusions was handled by using four overhead cameras. The knowledge of static activity zones in a kitchen environment was exploited in [Fleischman et al., 2006]; in a static overhead view of the kitchen, its furniture and large appliances were “traced over as regions of interest.” Motion history in the vicinity of these regions was mined for frequent patterns and a hierarchical representation of typical activities was computed. In [Breitenstein et al., 2008] a camera’s field of view captured an urban

outdoor scene. A formulation to recover 3D scene geometry from pedestrian detections was proposed. The approach also recovered a *walkability* map in the image plane and in 3D. In [Renno et al., 2007] motion patterns of commuters in a subway station were analyzed, and static occluding layers corresponding to the turnstiles were learned. The knowledge of these occluding layers improved tracking accuracy.

One limitation of all of these approaches stems from an assumption of a static scene. Indeed, in a parking-lot scenario, it may not be practical to learn activity zones around vehicles since these vehicles arrive and depart at random.

In order to recognize actions of people in dynamic environments, activity-grammar representations have been proposed. In [Moore et al., 1999] an object-centric model of appearance and action was proposed and formulated as the Generalized Class Model (GCM). Each GCM encapsulated an object’s appearance—e.g., a bounding box—and possible motion patterns near this object—e.g., writing motion. A scene was modelled as instances of GCM’s whose location and identity were inferred dynamically based on image evidence via a Bayesian Belief Network. Because of the advantageous viewing conditions, e.g., an overhead camera in the case of a working-at-a-desk scenario, the proposed formulation did not need to account for depth-ordering and inter-occlusions of GCM’s. In [Ivanov and Bobick, 2000] parking-lot interactions of pedestrians and vehicles were recognized by stochastic context-free grammars; one of the applications involved parking-lot surveillance. However, in the chosen camera viewpoint, high above the parking lot, few occlusions could be observed. Furthermore, it was not reported how such an approach could be extended to viewpoints in which occlusions are more severe. An approach to recognize person-vehicle interactions in the presence of occlusions was proposed in [Ryoo et al., 2010]. It adopted a Hidden Markov Model formulation, where the labels of hidden states included *approach the vehicle*, *open a door*, *close a door*, etc. To compute the state-likelihood, image evidence was compared against the rendered 3D articulated vehicle and person models; the depth-order required for rendering was based in part on 2D image coordinates of pedestrians’ feet. Experimental validation was conducted on video sequences of multiple passengers

interacting with a single vehicle.

Among the above approaches to modelling dynamic scenes, only [Moore et al., 1999] can be thought of as proposing an object-centric representation: GCM’s instantiated in the image plane. Thus, in principle this representation could be applied to scenes comprising numerous GCM’s. Indeed, experimental results [Moore et al., 1999] included scenarios with several model instances. However, as was mentioned earlier, due to the chosen camera viewpoint, issues related to mutual occlusions of the instantiated GCM’s were not addressed. Furthermore, interactions between GCM’s and the effect of these interactions on the computational complexity of activity-recognition algorithms were not specified.

## 2.4 Advantages to the Proposed Approach

We highlight key differences between our scene representation and prior work in Table 2.1. To make the comparison fair, we cite examples from prior work that are appropriate for person-tracking in non-overlapping views given static and/or relocatable occluders.

We consider four scene representations for tracking: image-plane regions [Smith et al., 2008, Yu et al., 2008, Yu and Medioni, 2009], depth-ordered layers [Jojic and Frey, 2001, Zhou and Tao, 2003, Titsias and Williams, 2006], explicit 3D [Dahlkamp et al., 2007, Ryoo et al., 2010, Leotta and Mundy, 2011], and layered graphical models defined in this thesis.

To summarize, our approach complements and extends the existing scene representations. Some of its benefits are as follows:

1. Our representation is derived from the layered paradigm. Therefore, we can take advantage of the existing approaches to learn some of the layer’s parameters. For example, 2D masks of relocatable occluders can, in principle, be learned using any existing layered formulation. Although mask-learning in a layered formulation may be too slow for a real-time application it is not a concern here, since in our representations 2D masks are acquired off-line.
2. Our representation is occluder-centric in the sense that it is developed for relocat-

**Table 2.1:** Comparison of scene representations for person-tracking in non-overlapping views in the presence of static and relocatable occluders.

Approach	Image-plane regions	Depth-ordered layers	Explicit 3D	Layered graphical models; implicit 3D
Examples	Smith2008 Yu2008 Yu2009	Jojic2001 Zhou2003 Titsias2006	Dahlkamp2007 Ryoo2010 Leotta2011	this thesis
Occluder masks	learned on-line	learned on-line	computed on-line	learned off-line; stored in a database
Persons' occlusions by masks	estimated during tracking	estimated during tracking	estimated during tracking	
Scene geometry	not modelled	not modelled	exact 3D→2D mapping	approximate 3D→2D mapping
Motion around occluders	not modelled	not modelled	volume-based constraints	Markov process on activity zones

able occluders and instantiated in the image plane where such relocatable occluders come to rest. Thus, our representation makes explicit the relations between people, whose tracks we wish to estimate, and the rest of the scene. As a consequence, the computational complexity of a person-tracking algorithm with respect to the scene can be derived in a straightforward fashion.

3. Our representation is *implicit 3D* in the sense that it models a person's realistic motion patterns in the ground plane and the corresponding image evidence. It therefore has the modelling capability of an explicit 3D representation but the advantage of relying solely on image-plane computations during tracking. One benefit is that the requirements of exact camera calibration during off-line database construction can be relaxed. Another benefit is that our scene representation can be maintained automatically when the image contrast and resolution may not be sufficient to enable

3D model-based alignment.

4. Our representation is efficient in the sense that it is instantiated only in the image regions that cover relocatable occluders at rest. Therefore, the computational complexity of tracking a variable number of pedestrians does not depend on the image evidence in the rest of the scene. Furthermore in a complete system our scene representation does not constrain the choices of free-space person-trackers.

## Chapter 3

# Scene Representation: Layered Graphical Models

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.

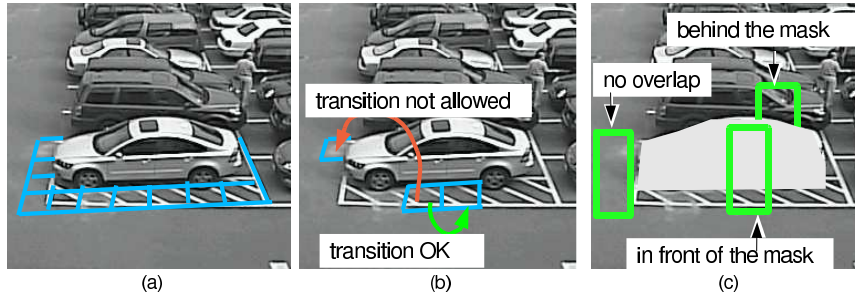
### 3.1 Basic Idea

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. 3-1.

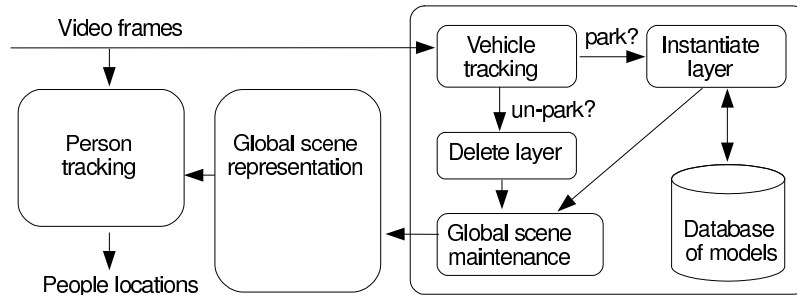
Because only the projection of motion in activity zones may be observed, our occluder-centric model is instantiated as a graphical model layer in the image plane. Expressing 3D mobility and visibility constraints implicitly in a layered framework opens the possibility of employing simpler image-plane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

In our global scene representation, a separate graphical-model layer is instantiated for every parked vehicle. For example, in Fig. 3-1, one layer is instantiated for the white sedan, another for 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. 3-2. Input video frames feed into a module that tracks vehicles as they arrive, park, or depart. When a vehicle parks, a pre-computed object-centric graphical



**Figure 3-1:** To convey the basic idea, we focus on one application, tracking of pedestrians in a parking lot, and define our object-centered representation for 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.



**Figure 3-2:** 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 pre-computed graphical-model layers. Please see the text for further details.

model is retrieved from the database of such models and instantiated as a layer in our global scene representation. Pre-computing a database of models is possible since the camera is fixed and a number of methods can be used to obtain ground plane calibration [Renno et al., 2002, Lv et al., 2006]. 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.

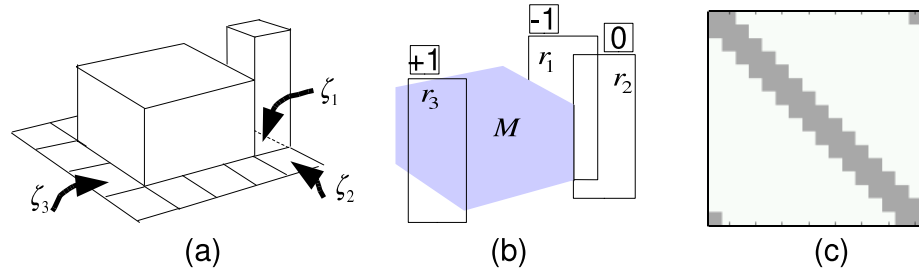
### 3.2 Local Representation: Graphical-Model Layer

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.

**Table 3.1:** Notation for graphical-model layer formulation.

$\zeta$	ground-plane <i>activity zone</i>
$Y$	a person’s state defined over $\zeta$ ’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 graphical-model layer formulation is summarized in Table 3.1. 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 non-overlapping squares, where each square is large enough to accommodate a person. In



**Figure 3-3:** (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  $\zeta_i$ 's. A person's motion on these zones is modelled as a first-order Markov process. For example, a stochastic transition from  $\zeta_1$  to  $\zeta_2$  is likely, while transition from  $\zeta_1$  to  $\zeta_3$  is not. (b) The occlusion mask  $M$  and a subset of  $R$  comprising three depth-ordered observation regions are shown. (c) The transition matrix for a model with sixteen 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.

Fig. 3-3a the squat box corresponds to a relocatable object. Three of its activity zones are labelled  $\zeta_1, \zeta_2, \zeta_3$ . 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. 3-3a it is reasonable to expect that the probability of a transition from  $\zeta_1$  to  $\zeta_2$  is high, while the probability of a transition from  $\zeta_1$  to  $\zeta_3$  is very low. These transition probabilities are summarized in a transition matrix; a transition matrix for a simple example model is illustrated in Fig. 3-3c.

A person standing in  $\zeta$  is approximated in the image plane by a bounding rectangle  $r$ , called an *observation region*. In our implementation observation regions correspond to a height 1.8 meters to fully cover persons of likely heights. Projection of the relocatable

object yields an *occlusion mask*  $M$ . Observation regions are depth-ordered with respect to  $M$ . Fig. 3-3b shows the occlusion mask corresponding to the relocatable object in Fig. 3-3a. 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 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.

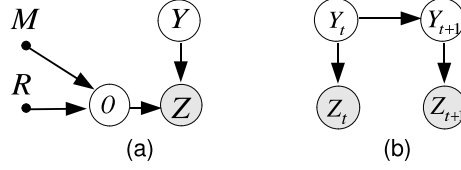
In many practical applications, image evidence is computed for every image location  $u$ . For example, moving-pixel-detection image, dense optical flow, and color, fall into this category and have been applied to tracking humans. We denote by  $z_u$  an observation at a pixel  $u$ , and let  $Z = \{z_u\}$  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 its variables made explicit is summarized in Fig. 3-4a. A Hidden Markov Model interpretation is given in Fig. 3-4b, where only the image evidence and activity zone nodes are shown.

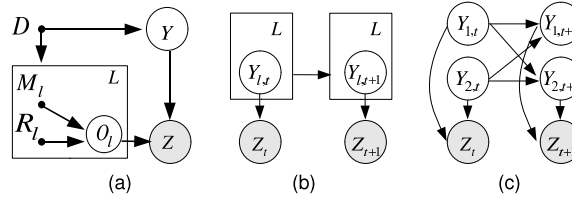
### 3.3 Global Scene Model: Depth-Ordered Layers of Graphical Models

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 im-



**Figure 3-4:** (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.



**Figure 3-5:** (a) In our global scene representation, comprising  $L$  layers, observations  $Z$  are generated by conditioning on a person’s location in the global state space  $Y$  and occlusion variables in all  $O_i$ ’s. 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  $Y_i$ ’s. Connections between  $Y_{i,t}$  and  $Y_{i,t+1}$  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  $p(Y_{2,t+1}|Y_{1,t}, Y_{2,t})$ .

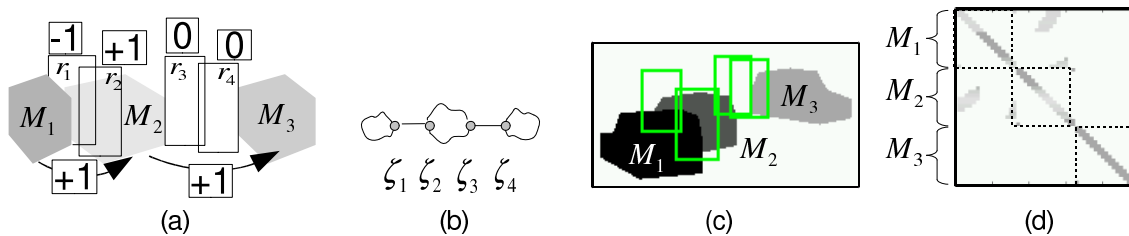
plementation we represent  $D$  as a set of variables, one for each pair of layers. The value of each variable is  $+1$  if the first layer occludes the second layer,  $-1$  if the second layer occludes the first one, and  $0$  if the two layers do not interact. We add a constraint on these variables 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  $Y_1, \dots, Y_L$ , with a constraint that any realization  $y_1, \dots, y_L$  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. 3-5a. The generative model for a single person moving around the relocatable occluders is summarized by Dynamic Bayes Net (DBN) in Fig. 3-5b, where dependence on  $D$  is not drawn to avoid cluttering the diagram. As an example, in Fig. 3-5c 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, which 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: (a) their observation regions overlap and are approximately the same size (b) these observation regions are not separated by an occlusion mask.

In the example in Fig. 3-6a 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



**Figure 3-6:** (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  $(\zeta_1, \zeta_2)$  and  $(\zeta_3, \zeta_4)$ . The edges added to the global transition graph are shown as straight lines. (c) A subset of a global scene from the PETS 2001 dataset, described in Sec. 6.3, with  $M_1, M_2, M_3$  corresponding to the rightmost three models of the bottom-left image in Fig. 6-1; 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.

$r_1$  is behind  $M_1$ ,  $r_2$  and  $r_3$  are respectively in-front of and non-overlapping  $M_2$ , and  $r_4$  is non-overlapping  $M_3$ . Given this configuration of observation regions and occlusion masks, our connectivity test is satisfied for the pairs of zones  $(\zeta_1, \zeta_2)$  and  $(\zeta_3, \zeta_4)$ . The added edges in the graphical model for the scene are shown as straight lines in Fig. 3-6b. In Fig. 3-6c we show instantiated occluder masks for a subset of real scene. The resulting global transition matrix is shown in Fig. 3-6d.

### 3.4 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 [Fleuret et al., 2008, Ge and Collins, 2009] demonstrate. Other fea-

**Table 3.2:** 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 $y_1, \dots, y_K$ 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

tures 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 3.2. Recall that the occlusions in each observation region  $r$  are modelled 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:

$$\tilde{o}_u = 1 - p(o_u = 0) = 1 - \prod_{l \in F_r} p(m_{l,u} = 0) = 1 - \prod_{l \in F_r} (1 - \tilde{m}_{l,u}), \quad (3.1)$$

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:

$$\begin{aligned}
p(z_u|y, R, M) &= \sum_{s_u} \sum_{o_u} p(z_u|s_u, o_u, y, R, M) p(s_u, o_u|y, R, M) \\
&= \sum_{s_u} \sum_{o_u} p(z_u|s_u, o_u) p(s_u|y) p(o_u|R, M) \\
&= \sum_{o_u} p(z_u|s_u = 1, o_u) p(o_u|R, M)
\end{aligned} \tag{3.2}$$

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

$$p(z_u = 1|y, R, M) = q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u) \tag{3.3}$$

and

$$p(z_u|y, R, M) = [q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u)]^{z_u} \times [1 - (q_2 \tilde{o}_u + q_1(1 - \tilde{o}_u))]^{1-z_u}. \tag{3.4}$$

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

$$p(z_r|y, R, M) = \prod_{u \in r} p(z_u|y, R, M). \tag{3.5}$$

For any pixel  $u \notin r_y$  we have

$$p(z_u|R, M) = (q_2)^{z_u} (1 - q_2)^{1-z_u}. \tag{3.6}$$

Lastly, 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  $y_1, \dots, y_K$ . 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  $K$  persons and their occlusion relations in the observation regions. If  $D^{\text{persons}}$  specifies the depth order and  $u$  is an image

pixel that belongs to the non-empty intersection of the observation regions corresponding to  $\mathbf{y} = (y_1, \dots, y_K)$  we can write

$$p(z_u | \mathbf{y}, D_{y_1, \dots, y_K}^{\text{persons}}; R, M) = p(z_u | \mathbf{y}; R, M). \quad (3.7)$$

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  $o_{u,k} \neq 1$

$$p(z_u = 0 | s_{u,1} = 1, \dots, s_{u,K} = 1, o_{u,1}, \dots, o_{u,K}) = 1 - q_1, \quad (3.8)$$

and it can be shown that

$$\begin{aligned} p(z_u = 0 | \mathbf{y}; R, M) &= \sum_{\mathbf{o}_u} p(z_u = 0 | s_{1,u} = 1, \dots, s_{K,u} = 1, o_{1,u}, \dots, o_{K,u}) \\ &\quad \times p(o_{1,u}, \dots, o_{K,u}; R, M) \\ &= (1 - q_2) \prod_k \tilde{o}_{k,u} + (1 - q_1) \sum_{\mathbf{o}_u \neq \mathbf{1}} \prod_k p(o_{k,u}). \end{aligned} \quad (3.9)$$

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

Although observation regions for several persons may intersect, our tracking algorithm will track these persons 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 [Fleuret et al., 2008] a person was approximated by a rectangle, and a generative model was developed to produce “ideal random images”. A pseudo-distance between those images and the actual binary observations was used in constructing the likelihood of a person’s location. In [Sigal and Black, 2006, Sigal, 2008] 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  $\Omega_1$ , those in its immediate vicinity  $\Omega_2$ , and the rest  $\Omega_3$ . 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 [Rasmussen and Hager, 2001]. 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.

## Chapter 4

# Tracking a Variable Number of People with Layered Graphical Models

Given the above layers-of-graphical-models representation, we now turn our attention to tracking a variable number of people around relocatable occluders. We are interested in developing tracking algorithms that may realize the full benefit of our scene representation in scenarios characterized by noisy image evidence and other sources of uncertainty. Therefore, in this chapter we define two tracking algorithms—one stochastic and one deterministic—that model uncertainty in the locations of people in the vicinity of the instantiated graphical-model layers.

The stochastic algorithm derived in this chapter is based on Reversible Jump Markov Chain Monte Carlo (RJ MCMC) and offers the advantage of explicitly representing uncertainty in a target’s location; it is implemented in a causal framework. The deterministic algorithm derived in this chapter is based on the Viterbi decoding of the trellis comprising the activity zones. The Viterbi-based algorithm offers the advantage of yielding a point estimate of a target’s location; it is implemented in a sliding-window framework.

Our experiments in subsequent chapters will demonstrate the effectiveness of the RJ MCMC-based and the Viterbi-based tracking algorithms for counting and localizing people using our scene representation. At the same time it will become evident from the experimental results that for scenarios such as parking-lot surveillance, which are characterized by low resolution and noise image evidence, the sliding-window Viterbi-based algorithm might be the preferred component of a complete system.

## 4.1 Tracking People with Reversible Jump Markov Chain Monte Carlo

Inferring  $y_t$ , the location of a target at time  $t$ , given  $\mathbf{Z}_{1:t}$ , the evidence in the last  $t$  image frames is a challenging problem. The challenge stems from our inability in most cases to evaluate in closed form multi-dimensional integrals involving  $p(\mathbf{Z}_t|y_t)$ , the state likelihood, and  $p(y_t|y_{t-1})$ , the state transition probability functions. One strategy to overcome this challenge is to approximate probability distributions by samples and the integrals involving such distributions by finite sums [Andrieu et al., 2003]; this strategy has been extended to target-tracking where the probability distributions vary with time [Doucet et al., 2001].

In scenarios where  $K$  targets are present in the scene, estimating the multi-target state  $\mathbf{Y}_t = \{y_k\}_{k=1}^K$  might become more challenging. One difficulty could stem from the interactions of targets' dynamics yielding complex dependencies between  $\mathbf{Y}_{t-1}$  and  $\mathbf{Y}_t$ . Such complex dependencies may be approximated by sampling. For example, when the approximation takes the form of Markov Chain Monte Carlo (MCMC), [Khan et al., 2004] proposed to sample from the *interaction factors*.

In some application domains  $K$  is unknown and varies with time. Therefore,  $K$  and  $p(\mathbf{Y}_t|\mathbf{Z}_{1:t})$  must be estimated for each  $t$ , making the exact inference impractical in many cases. One approach to approximate  $K$  and  $p(\mathbf{Y}_t|\mathbf{Z}_{1:t})$  jointly that might be practical from the standpoint of computational complexity is via RJ MCMC. In this formulation, the varying state-space dimensionality is handled by employing *birth* and *death* moves to hypothesize a new target or remove an existing one. Examples of tracking a variable number of targets in video sequences via RJ MCMC include [Khan et al., 2005] for ant colonies and [Smith et al., 2005a, Smith et al., 2008] for tracking multiple persons.

We now define the RJ MCMC tracking algorithm for scenes comprising the instantiated graphical-model layers; the notation is summarized in Table 4.1. As is common practice [Khan et al., 2005, Smith et al., 2008], the multi-target filtering distribution is expressed

**Table 4.1:** Notation for RJ MCMC-based tracking formulation.

$p(\mathbf{Z}_t \mathbf{Y}_t)$	observation likelihood at time $t$
$p_V(\mathbf{Y}_t \mathbf{Y}_{t-1})$	person dynamics; a random walk with given activity zone transition probabilities
$\{\mathbf{Y}_t^1, \dots, \mathbf{Y}_t^S\}$	chain of RJ MCMC sampled states at time $t$
$\mathbf{Y}_t^{n*}$	a proposed state at iteration $n$ of RJ MCMC chain for time $t$
$\mathbf{Y}_t^n$	accepted state at $n^{\text{th}}$ iteration
$i^*$	id of target chosen at an RJ MCMC iteration to apply one of the moves, $i^* \in \{1, \dots, K + 1\}$
$\alpha_b, \alpha_d, \alpha_u, \alpha_s$	acceptance probabilities for RJ MCMC moves
$p_v(\cdot)$	probability for sampling each of four moves
$p_V(\cdot \cdot)$	target dynamics
$\mathcal{N}_i$	neighbors of target $i$ in the interaction graph built on-the-fly
$\phi(y_i, y_j)$	interaction term to prevent collapse of multiple people states onto a single location
$q_b(\cdot), q_d(\cdot)$	constraints on activity zone locations where pedestrians can enter/exit the scene

as

$$p(\mathbf{Y}_t|\mathbf{Z}_{1:t}) = \mathcal{Z}^{-1} p(\mathbf{Z}_t|\mathbf{Y}_t) \int p(\mathbf{Y}_t|\mathbf{Y}_{t-1}) p(\mathbf{Y}_{t-1}|\mathbf{Z}_{1:t-1}) d\mathbf{Y}_{t-1}, \quad (4.1)$$

where  $\mathcal{Z}$  is the partition function. We approximate this distribution with  $S$  samples

$$p(\mathbf{Y}_t|\mathbf{Z}_{1:t}) \approx \mathcal{Z}^{-1} p(\mathbf{Z}_t|\mathbf{Y}_t) p_0(\mathbf{Y}_t) \sum_{n=1}^S p_V(\mathbf{Y}_t|\mathbf{Y}_{t-1}^{(n)}), \quad (4.2)$$

where each sample  $\mathbf{Y}_t^{(n)}$  defines a valid multi-person configuration on the activity zones of our instantiated scene model.

Samples from Eq. 4.2 are drawn via RJ MCMC with four move types: *birth*, *death*, *update*, *swap*. *Birth* changes the model order from  $K$  to  $K + 1$ , *Death* is its inverse, *Update* changes a target's position, and *Swap* swaps identities for a pair of targets.

In iteration  $n$  of the MCMC chain at time  $t$ , a state  $\mathbf{Y}'_t$  is chosen at random from the sample set at  $t - 1$ ,  $\mathbf{Y}'_t \sim \{\mathbf{Y}_{t-1}^1, \dots, \mathbf{Y}_{t-1}^S\}$ . A target  $i^*$  and move  $v$  are randomly chosen

and applied to  $\mathbf{Y}'_t$ , resulting in a proposed state  $\mathbf{Y}_t^{n*}$ . If the sample  $\mathbf{Y}_t^{n*}$  is accepted with probability  $\alpha_{(\cdot)}$ ,  $\mathbf{Y}_t^n = \mathbf{Y}_t^{n*}$ ; if rejected,  $\mathbf{Y}_t^n = \mathbf{Y}_t^{n-1}$ .

The *birth move*'s proposal distribution  $q_b(\cdot)$  keeps all current objects fixed and assigns non-zero probability to configurations containing a new target  $i^*$ . The interaction  $\phi(y_i, y_j)$  prevent states of multiple people from collapsing onto a single location. The acceptance ratio is

$$\begin{aligned}\alpha_b &= \min(1, R_b) \\ R_b &= \frac{p(\mathbf{Z}_t | \mathbf{Y}_t^{n*})}{p(\mathbf{Z}_t | \mathbf{Y}_t^{n-1})} \frac{\prod_{j \in \mathcal{N}_{i^*}} \phi(\mathbf{Y}_{i^*,t}^{n*}, \mathbf{Y}_{j,t}^{n*})}{1} \frac{p_v(\text{death})}{p_v(\text{birth})} \frac{q_d(i^*)}{q_b(i^*)}.\end{aligned}\quad (4.3)$$

The *death move*'s proposal distribution  $q_d(\cdot)$  assigns non-zero probability to configurations in which all objects are fixed and  $i^*$  has been removed. The acceptance ratio is

$$\begin{aligned}\alpha_d &= \min(1, R_d) \\ R_d &= \frac{p(\mathbf{Z}_t | \mathbf{Y}_t^{n*})}{p(\mathbf{Z}_t | \mathbf{Y}_t^{n-1})} \frac{1}{\prod_{j \in \mathcal{N}_{i^*}} \phi(\mathbf{Y}_{i^*,t}^{n-1*}, \mathbf{Y}_{j,t}^{n-1*})} \\ &\quad \times \frac{p_v(\text{birth})}{p_v(\text{death})} \frac{q_b(i^*)}{q_d(i^*)}.\end{aligned}\quad (4.4)$$

The *update move*'s proposal distribution incorporates target dynamics  $p_V(\cdot)$  for target  $i^*$  while all other targets fixed. The acceptance ratio is

$$\alpha_u = \min\left(1, \frac{p(\mathbf{Z}_t | \mathbf{Y}_t^{n*})}{p(\mathbf{Z}_t | \mathbf{Y}_t^{n-1})} \times \frac{\prod_{j \in \mathcal{N}_{i^*}} \phi(\mathbf{Y}_{i^*,t}^{n*}, \mathbf{Y}_{j,t}^{n*})}{\prod_{j \in \mathcal{N}_{i^*}} \phi(\mathbf{Y}_{i^*,t}^{n-1*}, \mathbf{Y}_{j,t}^{n-1*})}\right).\quad (4.5)$$

The *Swap move*'s proposal distribution swaps two targets' state values and histories, keeping the rest fixed. The acceptance ratio is

$$\alpha_s = \min\left(1, \frac{p(\mathbf{Z}_t | \mathbf{Y}_t^{n*})}{p(\mathbf{Z}_t | \mathbf{Y}_t^{n-1})}\right).\quad (4.6)$$

### 4.1.1 Computational Complexity

The computational complexity of an RJ MCMC-based person tracker with  $S$  samples in the Markov chain can be written as

$$O(S \cdot \sum_v p_v \cdot C_v), \quad v \in \{\text{birth, death, move, swap}\}, \quad (4.7)$$

where  $C_v$  is the computational complexity of evaluating each of the four moves. This equation is noteworthy not because of what it explicitly states, which follows from the axioms of probability, but because of what is left out.

The first dependency that appears to be missing from Eq. 4.7 is on  $K$ , the number of people in the scene. This may seem implausible since we would expect to require more samples in the Markov chain to approximate the posterior distribution as the number of people in the scene increases. Indeed, this intuition is borne out in practice, but the precise relation between  $K$  and  $S$  is not easily derived. Some of the difficulties associated with obtaining the theoretical bounds on the number of Markov chain samples required to obtain an independent sample from the posterior distribution are highlighted in [MacKay, 2003].

The second dependency that appears to be missing from Eq. 4.7 is that  $C_v$  is not written as a function of  $S$ . Indeed, one of the theoretical advantages of RJ MCMC is that the acceptance ratios may not require the evaluation of integrals or summations over the samples, since, for example, the partition function cancels out between the numerator and the denominator.

In extending RJ MCMC to our scene representation we can realize another advantage in terms of computational complexity. Since  $\mathbf{Y}_t$  is defined on activity zones which define a finite set, the evaluation of a function that depends only on  $\mathbf{Y}_t$  may be cached. In particular, the state likelihood function  $p(\mathbf{Z}_t | \mathbf{Y}_t)$  is a function of  $\mathbf{Y}_t$  only. Therefore when the Markov chain at time  $t$  is generated, the state likelihood values required by the acceptance tests can be efficiently cached in a data structure. In our experiments we have found that this

simple trick tends to speed up the throughput of our tracker severalfold.

It is evident that  $C_v$  would in general depend on the dimensionality of the image evidence. In our scene representation,  $C_v$  may be a function of the number of pixels in the union of all the observation regions. Because the observation regions tend to overlap, further speedups may be possible; we do not consider them in detail since the RJ MCMC-based tracker would not be a preferred match for our scene representation. However, some of these speedups will be realized for our Viterbi-based tracker, which we derive next.

## 4.2 Tracking People with the Viterbi Algorithm

In some applications, such as video-based surveillance, it might be necessary to summarize a tracker’s output in every frame as a *tracking answer*, comprising the number and the location of the people in the scene. To compute such an answer, a tracking algorithm that computes a probability distribution over the locations and possibly the number of people in the scene would need to be followed by a post-processing stage. In the case of our scene representation, such a post-processing stage might require computing an expectation with respect to the activity zones.

Thus, the motivations for developing an alternative to RJ MCMC tracking formulation include our interest in directly computing the tracking answer. In designing such a tracker, one of the benefits we realize is being able to consider image evidence in a window of  $T$  consecutive frames. During tracking, these temporal windows overlap to allow tracks established in the previous window to be continued. The details of our tracking algorithm are presented next.

**Formulation.** To handle a variable number of persons, 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, the virtual activity zone has no corresponding observation region. We summarize our notation for tracking in Table 4.2. Let  $\mathbf{Y}_t$  be the location of all people at time  $t$ , and let  $\mathbf{Y}_{1:T}$  be shorthand for  $\mathbf{Y}_1, \dots, \mathbf{Y}_T$

**Table 4.2:** Notation for Viterbi-based 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 people at time $t$
$\mathbf{Y}_{1:T}$	sequence $\mathbf{Y}_1, \dots, \mathbf{Y}_T$
$\mathcal{L}(y_{1:T}; \mathbf{Y}_{1:T}, Z_{1:T})$	log-likelihood of a track given other tracks $\mathbf{Y}_{1:T}$ and observation sequence $Z_{1:T}$

in the temporal window of  $T$  image frames. The quantity of interest is the posterior distribution  $p(\mathbf{Y}_{1:T}|Z_{1:T})$ , which is proportional to the likelihood of the multi-person state multiplied by the prior:

$$p(\mathbf{Y}_{1:T}|Z_{1:T}) \propto p(Z_{1:T}|\mathbf{Y}_{1:T})p(\mathbf{Y}_{1:T}). \quad (4.8)$$

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 [Fleuret et al., 2008], we estimate trajectories sequentially. Given a person's Markov process on activity zones, the probability of a single trajectory  $y_{1:T}$ , given the set of already-found trajectories  $\mathbf{Y}_{1:T}$ , and image evidence  $Z_{1:T}$ , can be written as

$$\begin{aligned} p(y_{1:T}|Z_{1:T}; \mathbf{Y}_{1:T}) &\propto p(Z_{1:T}|y_{1:T}; \mathbf{Y}_{1:T})p(y_{1:T}) \\ &= p(y_1)p(Z_1|y_1; \mathbf{Y}_1) \prod_{t=2}^T \underbrace{p(y_t|y_{t-1}; \mathbf{Y}_t)}_{\text{zone transition}} \underbrace{p(Z_t|y_t; \mathbf{Y}_t)}_{\text{image evidence}} \end{aligned} \quad (4.9)$$

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

$$\hat{y}_{1:T} = \arg \max_{y_{1:T}} p(y_1)p(Z_1|y_1; \mathbf{Y}_1) \prod_{t=2}^T p(y_t|y_{t-1}; \mathbf{Y}_t)p(Z_t|y_t; \mathbf{Y}_t) \quad (4.10)$$

can be found efficiently using the Viterbi algorithm.

As was mentioned in Sec. 3.4, 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 Eq. 4.10, 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 Eq. 4.10. 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

$$\mathcal{L}(y_{1:T}; \mathbf{Y}_{1:T}, Z_{1:T}) = \ell_{\text{init}}(y_1) + \ell_{\text{img}}(y_1; \mathbf{Y}_1, Z_1) + \sum_{t=2}^T \{\ell_{\text{trans}}(y_t; y_{t-1}, \mathbf{Y}_t) + \ell_{\text{img}}(y_t; \mathbf{Y}_t, Z_t)\}. \quad (4.11)$$

In Eq. 4.11 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  $\mathbf{c} = c_0, \dots, c_4$  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 re-learned 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 Eq. 4.11 our top-level tracking algorithm is straightforward and comprises 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.

#### 4.2.1 Computational Complexity

As shown in [Duda et al., 2001], 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  $\Omega$  is the set of all the pixels in all the observation regions, then the computational complexity of evaluating  $\ell_{\text{img}}$  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  $K$  persons over  $T$  frames on  $N$  activity zones is

$$O(K[ \underbrace{T \cdot N^2}_{\text{transitions}} + \underbrace{T \cdot N \cdot |\Omega|}_{\text{image evidence}} ]). \quad (4.12)$$

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

$$O(K[T \cdot N^2 + T \cdot (\underbrace{|\Omega|}_{\text{base likelihood}} + N \cdot \underbrace{|r_{\text{average}}|}_{\text{local evidence}})]), \quad (4.13)$$

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 Eq. 4.13. 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.

#### 4.2.2 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 Eq. 4.10. Uncertainty in the scene representation is handled by explicitly taking the probability of occlusion at each pixel of an observation region into account in Eq. 3.2. This probability of occlusion is based on “soft” occupancy of each layer’s occlusion mask in Eq. 3.1, 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 [Kschischang et al., 2001] to compute the marginal distributions over activity zones for every video frame.

### 4.3 Comparison of the RJ MCMC-based and the Viterbi-based Tracking Algorithms

In this chapter we have derived two algorithms for tracking a variable number of people in the vicinity of instantiated graphical-model layers. Both algorithms were able to realize the benefit of our scene representation by inferring the likely sequence of the activity zones using the image evidence in the observation regions.

The RJ MCMC-based algorithm offers several advantages. The uncertainty in the location of people in the scene is explicitly represented as samples in the Markov chain. The algorithm can cope with complex dependencies in the multi-target dynamics by evaluating pairwise potentials in the interaction graph built on-the-fly. The computational complexity of the algorithm is linear in the number of samples in the Markov chain. Computational speedups may be realized due to the finite cardinality of the state space and the typical overlap in the observation regions.

In some cases the RJ MCMC-based tracker might not be the preferred choice. For example, if an application requires that a sampling-based tracking algorithm provide a tracking answer, a post-processing step may be necessary. In the case of RJ MCMC derived for the activity zones, the post-processing step might require computing an expectation over the activity zones; if the samples in the Markov chain are not concentrated, computing the answer may not be straightforward. Another aspect of the RJ MCMC algorithm that needs to be noted is that the number of targets in the scene is computed deterministically. An approach to model uncertainty in the number of targets in a sampling formulation was demonstrated in [Sidenbladh and Wirkander, 2003] based on finite set statistics (FISST), but it is not clear if the increase in the computational complexity of FISST as the number of targets increases would make this approach practical.

The Viterbi-based algorithm also offers several advantages. The algorithm computes the tracking answer for the location and the number of the targets in the scene; thus no post-processing step is necessary. The algorithm is practical to apply in a temporal

window of video frames, which tends to improve accuracy; a sliding-window implementation allows the algorithm to run causally. The computational complexity of the Viterbi-based algorithm can be derived in a straightforward manner. The computational complexity of the algorithm may be reduced by re-using the computations in the Viterbi trellis.

In some cases the Viterbi-based tracking algorithm might not be applicable. For example, some video-analytics applications may require a probability distribution over the activity zones or over the tracks. Some of this information, such as marginal distributions for each activity zone, can be computed in a post-processing step, e.g., [Kschischang et al., 2001]. However, it might not be straightforward to obtain the joint distribution over all the quantities of interest, such as the number and the location of the complete tracks in a window of video frames using the output of the Viterbi-based tracker.

In the following two chapters we will demonstrate the effectiveness of our scene representation for tracking people in the vicinity of the relocatable occluders in the presence of occlusions. In order for us to quantitatively compare the output of our tracking algorithm with the state of the art, we require that the tracking algorithm compute the tracking answer for each video frame of a test sequence. Since the RJ MCMC-based algorithm requires a post-processing stage to compute such an answer and because deriving the tracking answer might not be straightforward in all cases, this tracking algorithm will be evaluated qualitatively. A quantitative comparison with the state of the art will be conducted using our Viterbi-based algorithm.

## Chapter 5

# Experiments with the RJ MCMC-based Tracker

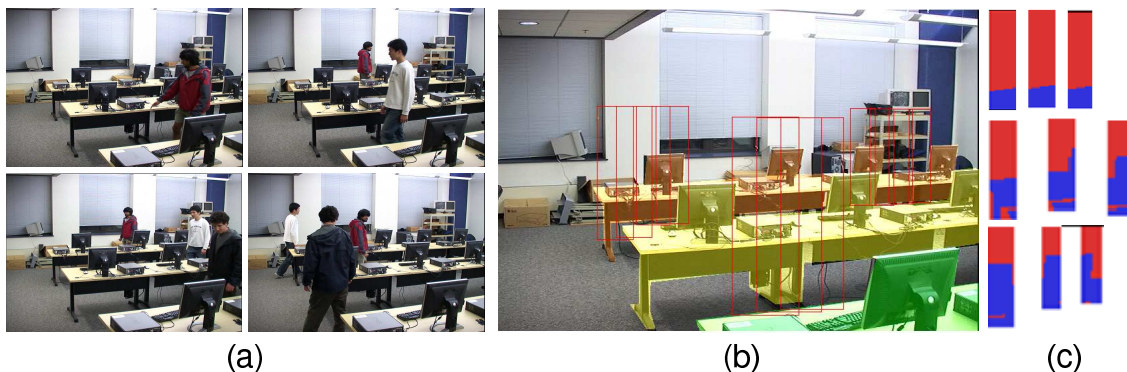
In this chapter we demonstrate the effectiveness of our scene representation in conjunction with the RJ MCMC tracker developed in Sec. 4.1. For our experiments we chose an indoor dataset captured in a computer laboratory and an outdoor dataset captured in a parking lot of an office building. As was mentioned in the previous chapter, the RJ MCMC tracker has a number of theoretical benefits, but it may not be the preferred choice in challenging scenarios characterized by low resolution and poor contrast. Nonetheless, qualitative assessment of the failure modes of the RJ MCMC tracker might provide some insights for its applicability to novel problem domains.

### 5.1 Datasets and Implementation Details

Before proceeding with the evaluation we describe the datasets and the implementation details of our RJ MCMC tracker.

**Computer Laboratory dataset.** This dataset was collected in a computer laboratory, and was presented in [Ablavsky et al., 2008]. In this dataset the scene comprises occluders—desks and computer monitors—that remain stationary throughout the video sequences. Persons walking in the laboratory become partially occluded when they step inside the aisles between the rows of desks. The video sequences are captured with a color Sony camcorder with resolution of 720 x 480 pixels, non-interlaced at 15 fields per second. A person of average height projects onto a rectangle of size 180 x 360 pixels in the near field, and of size 45 x 160 pixels in the far field.

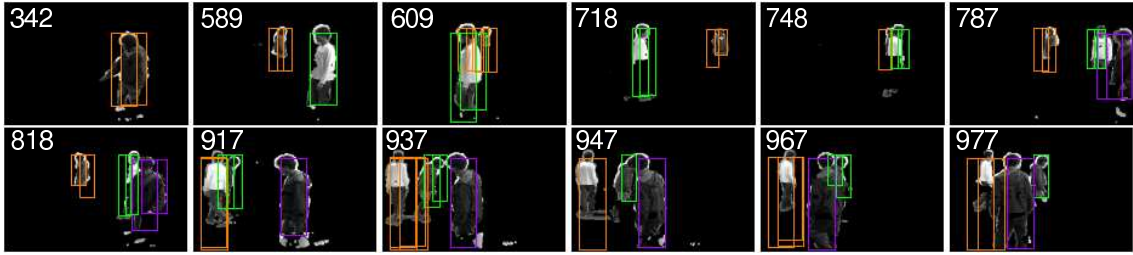
For this dataset we demonstrate that the benefit of our scene representation can be



**Figure 5-1:** Selected video frames from the test sequence of our Computer Laboratory dataset are shown in (a); observation regions selected from the three desk aisles are shown in (b); the occlusion variables for these observation regions are visualized as binary masks in (c) with blue (dark) indicating a high probability of occlusion.

realized with an approximate scene geometry, rather than an explicit 3D-to-2D mapping. Indeed, observation regions are specified by manually annotating bounding rectangles of a walking person in several video frames and computing the remaining observation regions via interpolation; our scene representation for this dataset comprises 64 observation regions. Since the computer monitors and the desks remain stationary it is practical to specify their static occlusion masks and their depth-order with respect to the observation regions. Example video frames and observation regions are shown in Fig. 5-1.

**Cambridge Office Park 2007 dataset.** This dataset and the corresponding database of graphical-model layers are described in detail in Chapter 6, which focuses on the evaluation of our Viterbi-based tracker. Briefly, video sequences in the COP2007 dataset were collected at an office park in Cambridge, MA, during the morning and evening peak hours. The image size in each of these videos is 720 x 480 pixels; the projected size of vehicles ranges from 170 x 70 up front to 54 x 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 x 18. To generate a database we define five relocatable object types—sedan, van, hatchback, station wagon, and mini-van—and for each type we define a coarse 3D polygonal mesh, and define a ring of square, non-overlapping activity zones around a vehicle. To demonstrate



**Figure 5.2:** This figure shows the tracking results for the Computer Laboratory test video sequence. Samples of  $p(\mathbf{Y}_t|Z_{1:t})$  in the Markov chain are visualized their corresponding observation regions, color-coded by each target’s id. The temporal index of each image frame is indicated in its upper-left corner.

the RJ MCMC-tracker we focus on the video sequences which do not contain examples of vehicles arriving and departing legal parking spaces; thus, an automatic scene-maintenance module is not implemented.

## 5.2 Qualitative Evaluation

We apply our RJ MCMC tracker to our two test datasets and compare the lifespan of the estimated tracks with the image-truth annotated by a human subject.

**Evaluation on the Computer Laboratory dataset.** For this dataset, parameters of our tracker are specified as follows. The probabilities of the birth, death, update moves are set to 0.1, 0.01, 0.09 respectively; the probability of the swap move is set to 0 since a person’s texture is not explicitly represented by our generative model. The interaction potential for a pair of activity zones is modelled as  $\phi(\cdot, \cdot) \propto \exp(-\lambda\mu)$ , with  $\mu$  equal to the overlap between the observation regions corresponding to the two activity zones. We set  $\lambda = 100$  for a pair of activity zones owned by the same layer and  $\lambda = 0$  for a pair of activity zones owned by different layers. A total of forty samples are drawn in an RJ MCMC chain at each time step; as is standard practice, e.g., [Smith et al., 2005b, Khan et al., 2005], the first 25% are regarded as burn-in samples.

We run our RJ-MCMC tracker on our scene representation for the Computer Laboratory test sequence. The tracker’s output is compared with the image-truth specified by a

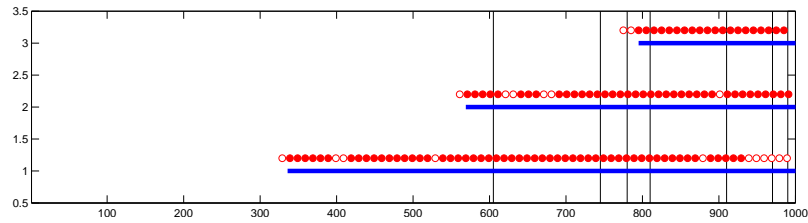
human subject. The image-truth comprises a temporal lifespan for each target.

The initial video frames of our test sequence contain no people in the scene, and our tracker correctly explains image evidence in the observation regions with zero people. As the first person enters the scene from the right at  $t = 330$ , our tracker correctly switches to a one-person state-space configuration via the birth move; the temporal discrepancy between the time of the first track’s initialization and the image truth is approximately ten frames, i.e., less than one second. The second person enters the scene at  $t = 550$ , and our tracker correctly switches to the two-person configuration via the birth move; the temporal discrepancy with the image truth is approximately ten frames. The third person enters the scene at approximately  $t = 760$ , and our tracker correctly switches to the three-person state-space configuration; the temporal discrepancy with the image truth is approximately twenty frames, less than two seconds. In all three cases our tracker accepts the birth move slightly before the image truth.

We visualize the output of our tracker in Fig. 5·2. In this visualization,  $p(\mathbf{Y}_t|Z_{1:t})$ , the posterior distribution for the multi-person state space with respect to activity zones is shown as the bounding rectangles of the observation regions. As Fig. 5·2 shows, for the Computer Laboratory test sequence, the Markov-chain samples for first person are concentrated in the correct activity zones. Indeed, since the observation regions partition the image plane somewhat coarsely, it is to be expected that a sampling-based tracker such as ours would represent a person’s state as samples over several adjacent activity zones.

As Fig. 5·2 shows, the samples in our Markov chain are concentrated in the activity zones whose observation regions are well-localized to the image evidence. Indeed, when the second person enters the scene and subsequently overlaps with the first person in the image plane, the Markov-chain samples from the posterior distribution correctly follow the two targets through occlusion. With the appearance of the third person, the posterior distribution remains concentrated on all three targets in spite of occlusions by each other and also the desks and the computer workstations.

In Fig. 5·3 we visualize the number of the targets and their lifespan as estimated by our



**Figure 5-3:** Tracking result timelines for the Computer Laboratory sequence whose sample frames are shown in Fig. 5-2. The vertical axis gives the track id and horizontal axis the frame numbers. The horizontal blue (dark) segments are ground-truth start and end for person tracks. The circles in red (light gray) are tracking results for our approach. The first person enters the scene shortly after frame 300, the second person enters the scene shortly before frame 600, and the third person enters the scene shortly before frame 800. The temporal difference between the frame index a person becomes fully-visible and the frame index at which our system creates the corresponding track, tends to be less than two seconds. Solid circles indicate that the person was hypothesized in the correct layer, and open circles imply an incorrect layer. Vertical lines mark instances of inter-person occlusions.

RJ MCMC-based tracker. The lifespan of each estimated track is visualized as a sequence of circles, one for each ten frames of the test video sequence. The image-truth lifespan for each of the three targets is shown as a horizontal solid blue line. As was mentioned earlier, the start of the image-truth tracks is less than two seconds apart from the start of our system’s tracks.

We also visualize the association of a person with the correct occlusion layer. A filled red (light-gray) circle indicates that our tracker made the correct association. For the majority of the time frames the association is correct. One source of misassociation seems to be occur at the start of each track. This may be due to targets entering the scene from the right-hand-side of the image and being occluded by both the computer monitor and the edge of the image.

In the parking-lot surveillance domains characterized by low contrast and low signal-to-noise ratio, the RJ MCMC-based tracker might not be the preferred match to our scene representation. Therefore, in the remainder of this chapter we focus on the evaluation of the Viterbi-based tracker.

**Evaluation on the COP2007 dataset.** The evaluation of our RJ MCMC-based tracker focused on two video sequences, called MINI Cooper and “far-field.” In the MINI Cooper video sequence, two pedestrians initially appear unoccluded near rows of parked vehicles, then proceed to walk between the vehicles toward a MINI Cooper. In the “far-field” video sequence, the capability of our scene representation to handle low image resolution is demonstrated; the pedestrians emerging from vehicles are less than twenty-five pixels tall.

We choose a subset of 350 frames from the MINI Cooper sequence, with two persons walking between the vehicles, to run our proposed approach. Probabilities for birth, death, and update moves were set to  $[0.0001, 0.0001, 0.9999]$  as within this sequence the number of persons to track is constant. In Fig. 5-4 frame 1171, we initialized the tracker to track two pedestrians at locations indicated by green arrows. A subset of the observation regions that overlap with pedestrians in the first frame are chosen as initial particles. In frame 1221, despite severe occlusion the particle set concentrates around the true location. In frame 1386 uncertainty in depth for one pedestrian causes particles (dark blue) to diffuse a bit. In frame 1514, particles are correctly concentrated on both sides of the MINI Cooper. We do not know of any other system that can correctly place two closely-spaced targets on both sides of an occluding layer in such a setting.

For the “far-field” sequence, we use 350 frames with interesting activity as shown in Fig. 5-5. A vehicle (indicated by the orange arrow) passes as pedestrians (green arrows) get out of parked cars. Birth, death, and update move probabilities are set to  $[0.001, 0.0005, 0.9985]$  so the system creates and deleted tracks on its own. A person in the farthest car remains seated after opening the driver-side door; his track hence latches onto the moving car and terminates as the car moves away from the observation regions. The person exits his car eventually, and his track is picked up and continues until the end of the sequence. The driver of the mini-van in the mid-field walks around his vehicle as the car passes. His track is not lost, because the activity zones corresponding to the observation regions activated by the car do not have transitions from the persons location. Instead, a new short track is created. As in the case of the MINI Cooper sequence, inferred activity zones



**Figure 5.4:** Tracking results for the MINI Cooper test video sequence of the COP2007 dataset are visualized as samples of  $p(\mathbf{Y}_t|Z_{1:t})$  in the Markov chain. Arrows point to the image-truth locations of the two pedestrians in the scene.



**Figure 5-5:** Tracking results for the “far-field” test video sequence from the COP2007 dataset are visualized as samples of  $p(\mathbf{Y}_t|Z_{1:t})$  in the Markov chain. Green (light) arrows point to the image-truth locations of the two pedestrians in the scene; orange (dark) arrows point to the moving vehicle.

reveal proximity to driver-side door and the passenger-side door.

Experiments with the three video sequences demonstrate the effectiveness of our scene representation for tracking partially-occluding moving persons. Our C++ tracking code runs at 4fps for the Computer Laboratory video and at almost video-frame-rate for the outdoor sequences, making it practical for real-time applications. In the case of the Computer Laboratory dataset, our tracking algorithm was able to accurately estimate the number and the location of people in the scene; the multi-person posterior distribution remained concentrated around the correct locations throughout occlusions. In the case of the COP2007 dataset the RJ MCMC-based tracking algorithm can automatically detect new people emerging from vehicles and track pedestrians through severe occlusions.

However, we have also found that due to the extremely low resolution and poor contrast, the samples tended to spread across the neighboring activity zones a bit. In some cases when the death proposal was accepted but according to the RJ MCMC formulation the target's samples remained in the Markov chain, they tended to diffuse over the activity zones. In principle, such diffusion could be remedied in a post-processing step common in many tracking formulations, [Xing et al., 2009, Yu and Medioni, 2009]. In applications such as parking-lot surveillance that require a point estimate of the targets' locations, it might be desirable to try a different algorithm. One such algorithm is our Viterbi-based tracker, and we evaluate its performance in the next chapter.

## Chapter 6

# Experiments with the Viterbi-based Tracker

We demonstrate our formulation in tracking people in the domain of parking-lot surveillance. As was mentioned in Chapter 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 non-overlapping view.

### 6.1 Scene Update Module

In order to apply layered graphical models to the parking-lot surveillance domain, it becomes necessary to instantiate and un-instantiate graphical-model layers for the vehicles that come to rest or depart the legal parking spaces. Fortunately, the design of our scene representation tends not to favor any particular scene-maintenance algorithm. Furthermore, since our formulation includes a database of graphical-model layers, a scene may be rapidly instantiated according to the specified parameters. Therefore, the design of such a scene-maintenance algorithm may be motivated by application-specific considerations, such as real-time requirements.

In principle, a scene-maintenance module might interact with other modules of a complete system, e.g., the pedestrian tracker. In addition, the scene-maintenance module might realize the benefit of accumulating the image evidence after a graphical-model layer has been instantiated. These considerations are re-visited in Chapter 7, which is a guide to

applying our formulation to new problem domains. For the sake of demonstrating our formulation, our scene-maintenance module is designed to operate independently of other modules in the system and does not modify the parameters of a graphical-model layer after its instantiation.

One approach to instantiating and un-instantiating graphical-model layers is to follow each vehicle in the scene with a separate vehicle-tracker and to detect when a vehicle has come to rest. Approaches to tracking vehicles using 3D models or 2D masks have been proposed by [Pece, 2006, Dahlkamp et al., 2007, Atev and Papanikolopoulos, 2008, Venkataraman et al., 2008]. In a recent work by [Leotta and Mundy, 2011], a parameterized 3D model was developed in a such a way that it could be adapted to the makes and models of typical passenger vehicles sold in the U.S. The experiments reported by [Leotta and Mundy, 2011] using HDTV-quality video sequences—1280 x 720 pixels at 30 frames per second, progressive scan—demonstrated the accuracy of tracking a single vehicle with such parameterizations and the ability to do so during short-term occlusions. However, it remains unclear, if this and similarly-expressive vehicle models may be applicable to lower-resolution video sequences and to scenarios in which a vehicle’s occlusions tend to be prolonged and severe. Although 2D patch-based appearance modelling methods were proposed by [Yin and Collins, 2007, Han and Davis, 2009], they might be confounded by the abrupt appearance changes that occur when a vehicle maneuvers into a parking spot and around already-parked vehicles.

In our implementation, the scene-maintenance module accumulates image evidence in the region corresponding to the legal parking spaces. The number, the location, and the image-plane orientation of the parking spaces are not provided as input to our system. However, in a system engineered for a specific scenario, such detailed information about the parking spaces may be specified during initialization. Furthermore, if a real-time, multi-vehicle tracking algorithm is found to be applicable, the output of such an algorithm might provide informative priors to our scene-maintenance module.

**Image evidence.** The image evidence used by our scene-maintenance module includes

sparse optical flow and moving-foreground connected components. Sparse optical flow of [Shi and Tomasi, 1994] is computed for every pair of adjacent video frames and yields the motion of small textured patches. The connected components or *blobs*, are computed for the image comprising moving pixels in each video frame; the moving pixels are detected by a background-subtraction algorithm.

In our implementation, the moving pixels are identified by the same background-subtraction algorithm as is used in our pedestrian tracker, but the use of the same background-subtraction algorithm for these two modules is not a requirement. Indeed, in our experiments we have found it advantageous to specialize the parameters of our background-subtraction algorithm to rapidly adapt to vehicles arriving and departing legal parking spaces. The reader is reminded that the image evidence used by the pedestrian tracker takes the form of the binary moving-pixel image computed by a background-subtraction algorithm without further post-processing.

These two types of image evidence—sparse optical flow and the foreground blobs—are accumulated in a temporal window. In principle, the temporal extent for inferring the scene may be chosen to suit the application requirements, e.g., to achieve the desired trade-off between the accuracy and the throughput, and may be based on imaging conditions, e.g., the camera’s frame rate. In our implementation, the sliding window’s temporal extent is taken to be the same as for the Viterbi-based tracking module, making it simpler to combine the outputs of the two modules.

**Layer instantiation.** Our algorithm for instantiating a graphical-model layer considers two competing hypotheses: one hypothesis explains the image evidence by a vehicle arriving into a parking region, and another hypothesis regards the image evidence as noise. To efficiently evaluate these hypotheses, our algorithm includes a bottom-up stage and a top-down stage.

The bottom-up stage is activated when a large foreground blob overlaps the parking region in the image plane. The foreground blob is then compared against the likely vehicle masks in the database indexed by the blob’s centroid. If the blob’s overlap with any vehicle

mask passes our coverage test, a distribution with respect to the vehicle occlusion mask’s orientation is estimated. This distribution is integrated over the entire temporal window, and then the likely occlusion mask orientation is determined.

The top-down stage is activated after a foreground blob has passed the coverage test with some vehicle mask in the database, and the likely orientation of the occlusion mask has been estimated. This occlusion mask found in the bottom-up stage, is employed to account for the image evidence in the entire temporal window, yielding a likelihood. If this likelihood reaches over a pre-defined threshold, the instantiating system decides that a vehicle is entering a parking spot. In the subsequent temporal windows the same mask is used to follow the vehicle until the image evidence from sparse optical flow indicates that the vehicle has come to rest. A graphical-model layer corresponding to this vehicle is then instantiated.

In our test video sequences the spatial extent of some arriving vehicles reaches only  $64 \times 22$  pixels. When such vehicles come to rest in front of the already-parked vehicles, the image evidence from sparse optical flow and moving blobs remains informative for our algorithm to infer the image-plane orientation and location of their occlusion masks. One way to extend our layer-instantiation algorithm to cope with occlusions of vehicles at such low resolution might be via stronger domain priors, e.g., utilizing the known location and the orientation of the parking regions. Extensions to our scene-update module under challenging scenarios without resorting to strong domain priors might lead to novel ways of interpreting the image evidence and would serve as an interesting direction for future work.

**Layer un-instantiation.** We also implemented an algorithm to un-instantiate layers from the global scene model. This algorithm relies on the same image evidence used for layer instantiation, but the steps are “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 sparse 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 un-instantiation module is discriminative enough to detect when a far-away vehicle “un-parks” while not being distracted by pedestrians walking past vehicles.

**Initialization.** Our complete parking-lot system is designed to start with a scene that is initially empty of parked vehicles. However, in each of our challenging video sequences, the first video frame captures a scene in which multiple parked vehicles are already present. Therefore 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 [Winn and Shotton, 2006, Wu and Nevatia, 2007, Arie-Nachimson and Basri, 2009] may be employed to segment layer masks in key frames. Such segmentation, if available, might serve as an independent source of information for the scene-maintenance module.

**Joint inference about the pedestrians and vehicles.** As the experiments in this section will demonstrate, state-of-the-art pedestrian detectors might not always be a good match for our challenging datasets. Therefore, we have to rely on cues compatible with the available image resolution and contrast, such as a blob’s 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 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.

## 6.2 Implementation Details

**Database of graphical-model layers.** For the parking-lot surveillance domain we define five relocatable object types—sedan, van, hatchback, station wagon, and mini-van—and for each type we define a coarse 3D polygonal mesh. We define a ring of square non-overlapping activity zones around a vehicle. For the hatchback, the smallest vehicle type, this yields sixteen activity zones, and for the remaining types it yields eighteen activity zones. Zone transition probabilities are defined so as to make transitions to immediate neighbors equally likely and to disallow jumps to non-neighboring zones; the same rule applies to the global scene representation as defined in Sec. 3.3.

We construct a database of models by deterministically sampling vehicle poses in the ground plane. We calibrate the camera using the approach of [Lv et al., 2006]. For each vehicle type, its orientation is sampled at sixteen 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 datasets, 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.

**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 [Bradski and Kaebler, 2008] with default parameters, except for `bg_threshold=0.9`. That value was chosen manually to make the background model

rapidly adapt to parking and un-parking vehicles on the “training” sequence from the PETS 2001 dataset 1, camera view 1.

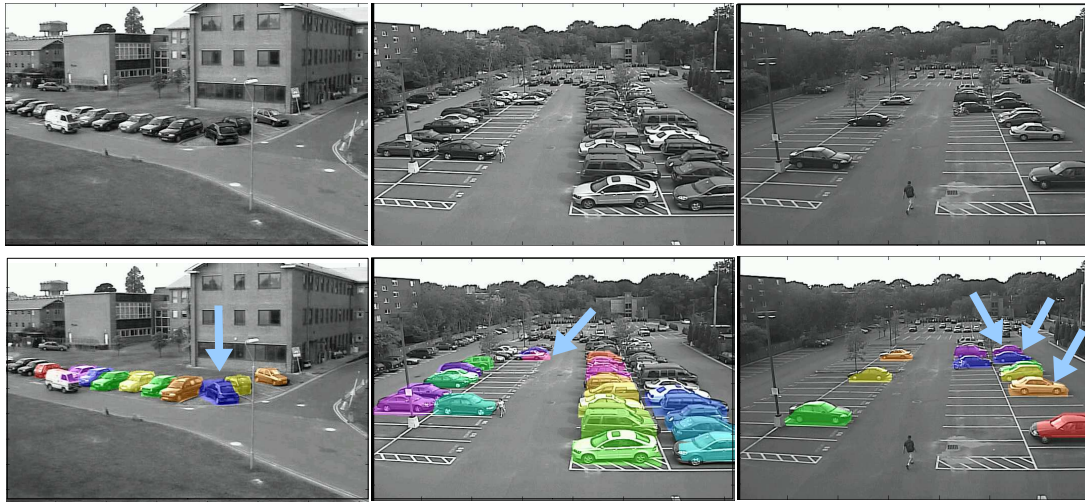
- To model  $Z$  we set  $q_1 = 0.6$  and  $q_2 = 0.1$ .
- To generate optical flow we use an implementation of [Birchfield, 2007] with default parameters.
- The maximum number of pedestrians to track simultaneously is set to 10.
- Since the PETS2001 “training” sequence does not have ground-truth bounding boxes, our Viterbi-based 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.

### 6.3 Datasets

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. 6.1. The six parking-lot video sequences are summarized in Table 6.1. We next describe our test data in greater detail.

**PETS 2001 dataset.** This dataset was originally presented at PETS 2001 [pets, 2001] and has served as a benchmark for numerous studies, e.g., [Siebel and Maybank, 2001, Senior et al., 2006, Zhu et al., 2008]. We focus on the “testing” sequence from the dataset 1, camera view 1, since occlusions in that view tend to be more severe. The size of each image frame is 768 x 576 pixels. The bounding box for the nearest vehicle, which happens to be facing away from the camera, is 66 x 46 pixels. The bounding box for an unoccluded pedestrian standing next to this vehicle is 15 x 44 pixels.

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 twenty ground-plane locations from this high-trafficked area, and for each



**Figure 6.1:** 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 un-instantiated later in this video sequence.

location generated one model for each of sixteen 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 dataset.** These sequences were collected at an office park in Cambridge, MA during the morning and evening peak hours, and were presented in [Gutchess et al., 2007]. 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 x 480 pixels. The projected size of vehicles ranges from 170 x 70 up front to 54 x 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 x 18. In our experiments, we focus on the middle and front portions of the lot.

A single database of models was shared among all the COP2007 sequences since they all

**Table 6.1:** 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

had been captured with the same camera parameters. Because sedans and station wagons in the U.S. tend to be larger than their European counterparts, we enlarged our coarse 3D models for these 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 dataset.

#### 6.4 Qualitative Evaluation

Before conducting the quantitative evaluation of our implementation, we first perform qualitative comparisons with two published methods [Siebel and Maybank, 2001, Titsias, 2005]. The method of [Siebel and Maybank, 2001] employs a deformable-contour model, and qualitative results were published for test sequence (e), the sequence from the PETS 2001 dataset. The method of [Titsias, 2005] 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 the work of [Siebel and Maybank, 2001] a B-spline contour was fit to pedestrian-sized foreground blobs and

projected onto a learnt 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 modelled 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 [Siebel and Maybank, 2001] this system is applied to sequence (e) from the PETS 2001 dataset, 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 1036 and 1147 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 1213 indicates that some of these individuals are tracked, but it is not clear exactly what happens during the period of occlusions. Since [Siebel and Maybank, 2001] 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. 6-2, 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 1089. Although no free-space vehicle-tracking is performed, the instantiated layer’s location and orientation is comparable to Fig. 9 of [Pece, 2006]; that system uses a 3D model-based ground-plane vehicle tracker with six



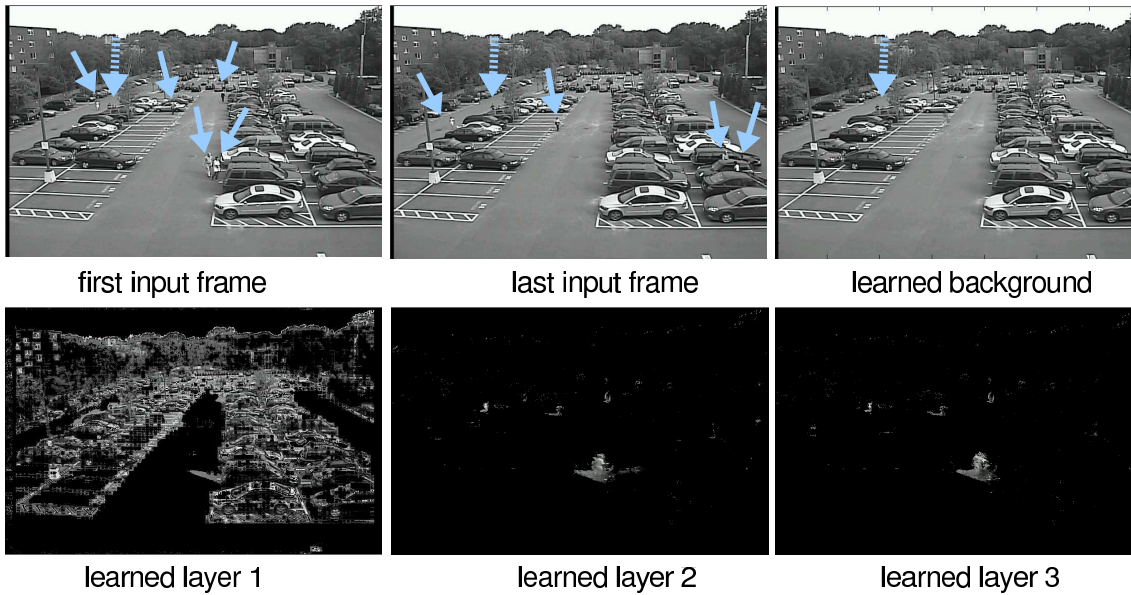
**Figure 6-2:** Example frames from our tracking algorithm applied to test video (a), top row, and to test video (e), the PETS 2001 sequence, bottom row. Rectangles indicate observation regions corresponding to the ground-plane zones selected by the tracker; the color of these rectangles in each frame is chosen for visual contrast. A missed pedestrian is marked with a dashed arrow; correct detections are marked with solid arrows.

degrees of freedom.

While it may seem that both the method of [Siebel and Maybank, 2001] 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 [Titsias, 2005]. 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 top-center images of Fig. 6-3, with arrows highlighting the pedestrians’ positions.

We use a publicly-available implementation of [Titsias, 2005] with default parameters,



**Figure 6-3:** 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.

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: to process our subsequence it requires 4.6 hours on Intel Core2 Quad 2.8GHz CPU.

In the top-right of Fig. 6-3 we show the background layer learned by [Titsias, 2005]. 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. 6-2.

The bottom row of Fig. 6-3 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 [Titsias, 2005]. 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 [Ablavsky et al., 2008] a qualitative comparison with a color- and texture-based multi-target tracker was performed using the implementation provided by [Takala and Pietikainen, 2007]. 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.

## 6.5 Quantitative Evaluation

Before presenting 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 un-instantiated occlusion masks. To enable such evaluation, two human subjects not involved in algorithm development provided spatio-temporal 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 [Smith et al., 2005b] as  $F = \frac{2\rho\nu}{\rho+\nu}$ , 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 6.2, 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 F-measure is 1.0, the performance is good; it also agrees with the examples shown in Fig. 6-1, where arrows point to the automatically-instantiated or un-instantiated layers.

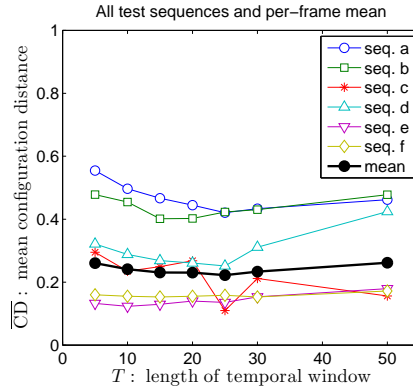
Vehicles entering a parking spot typically decelerate gradually before coming to rest. Our scene update module has to decide when the vehicle has stopped, which is complicated by low resolution and the fact the a vehicle may be “creeping”; hence there seems to be a bias in our system to instantiate models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

**Table 6.2:** 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

**Evaluation of pedestrian-tracking with changing scene models.** Although the PETS2001 dataset has served as a benchmark for numerous studies e.g., [Siebel and Maybank, 2001, Jepson et al., 2002, Senior et al., 2006, Zhu et al., 2008], there tends to be large variability in evaluation protocols. We adapt the evaluation metrics proposed in [Smith et al., 2005b] and used in [Smith et al., 2008] that are specifically designed for multi-object tracking systems.

As an overall performance measure we employ the configuration distance  $CD^t$  at time



**Figure 6.4:**  $\overline{CD}$  as a function of  $T$  on the complete system comprising a person-tracker and an automatic scene update module. Curves are shown for each of the six parking lot video sequences listed in Table 6.1. The mean performance curve in the graph is computed using the  $\overline{CD}$  over all frames in the six sequences.

$t$  and the average configuration distance  $\overline{CD}$ , computed over a video sequence of length  $n$ :

$$CD^t = \frac{N_{\mathcal{E}}^t - N_{\mathcal{GT}}^t}{\max(N_{\mathcal{GT}}^t, 1)}, \quad \overline{CD} = \frac{1}{n} \sum_{t=1}^n |CD^t|. \quad (6.1)$$

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

We assess the  $\overline{CD}$  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 6.1. The results of this assessment are shown in the graph of Fig. 6.4: as  $T$  increases the  $\overline{CD}$  tends to monotonically decrease to the global minimum and then it monotonically increases somewhat.

In [Fleuret et al., 2008] 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 un-parking 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, 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 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 [Wu and Nevatia, 2009] described in Sec. 2.2 was performed; it was done by the authors of [Wu and Nevatia, 2009] themselves. However, they 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 [Wu and Nevatia, 2009] parallels the observations reported in two other tracking-by-detection approaches. In [Xing et al., 2009] “...people that are too small in the images (less than 24 pixels in width) are not counted in the evaluation,” and in [Leibe et al., 2008] “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 [Leibe et al., 2008], described in Sec. 2.2, and the Latent-SVM (LSVM) pedestrian detector of [Felzenszwalb et al., 2010]. 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.

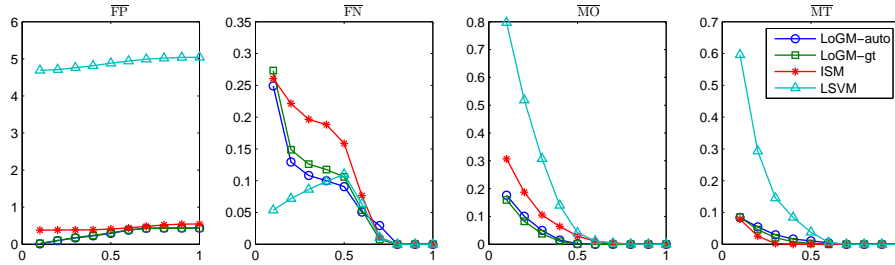
For such a comparison, the configuration distance (CD) alone may not provide enough insight into a tracker’s performance. Therefore, we adopt a more comprehensive set of

**Table 6.3:** Summary of the extended performance measures proposed in [Smith et al., 2005b].

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.

performance measures which were originally proposed in [Smith et al., 2005b] and applied to evaluate a tracking system in [Smith et al., 2008]. These additional performance measures are summarized in Table 6.3, 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 *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 non-negative numbers:  $\overline{FP}$ ,  $\overline{FN}$ ,  $\overline{MO}$ , and  $\overline{MT}$ . One shortcoming of these performance measures is that in the general case their average need not equal  $\overline{CD}$ .

As mentioned in Sec. 1.2, 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 [Leibe et al., 2008, Felzenszwalb et al., 2010] to work, each video frame of our test sequences must be up-sampled and



**Figure 6-5:** For test sequence (e), average configuration error measures from [Smith et al., 2005b] are plotted against the coverage-test threshold  $\tau_c$ . As  $\tau_c$  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 [Leibe et al., 2008], and Latent-SVM pedestrian detector of [Felzenszwalb et al., 2010].

interpolated by a factor of 2.5.

We apply our evaluation protocol to test sequence (e) from the PETS2001 dataset, with a set of coverage test thresholds  $\tau_c \in [0.1, 1]$ . As Fig. 6-5 indicates, overall, our systems based on layers of graphical models fare well, performing equally-well or better on all four performance measures. The LSVM approach of [Felzenszwalb et al., 2010] does not seem to be competitive on this dataset. With respect to the ISM approach, our systems achieve uniformly better results with respect to  $\overline{FP}$  as well as  $\overline{MO}$ , are competitive in terms of  $\overline{MT}$ , and are decisively better in terms of  $\overline{FN}$ . Indeed, for  $\tau_c = 0.5$  our system reduces ISM’s  $\overline{FP}$  by 29%,  $\overline{FN}$  by 42%, and  $\overline{MO}$  by 94%; for  $\tau_c = 0.5$   $\overline{MT}$  equals zero for our system and the ISM approach.

Comparing our system to a system comprising 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  $\overline{FN}$  for  $\tau_c < 0.5$ , the two systems are quite close in terms of performance everywhere else.

By varying  $\tau_c$  we change the criterion for a match between an estimated and a ground-truth bounding box. As  $\tau_c$  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  $\tau_c$  increases so does the mean false positive,  $\overline{\text{FP}}$ , as an increased number of estimated tracks do not match the ground-truth tracks. Because our evaluation protocol ignores false negatives (FN) originating from ground-truth tracks that do not pass the coverage test with at least one observation region of our scene model,  $\overline{\text{FN}}$  will tend to decrease as  $\tau_c$  increases. The multiple objects (MO) 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  $\tau_c$  increases  $\overline{\text{MT}}$  decreases for the same reasons as does  $\overline{\text{MO}}$ .

In summary, the tracking-by-detection approach of [Wu and Nevatia, 2009] was not a good match for this dataset, and performance of the LSVM pedestrian detector of [Felzenszwalb et al., 2010] was not competitive. Our pedestrian-tracker based on layers of graphical models with automatic scene update performed as well or decisively better than the ISM pedestrian detector of [Leibe et al., 2008] on all performance measures.

**Throughput.** Our video manipulation sub-system 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 2.83GHz Intel Core2 Quad CPU under Windows 2003 Server OS. Our person-tracker in isolation runs on average at 6.74Hz on the PETS2001 video sequence cropped to 720x480 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 [Wu and Nevatia, 2009] that their tracking-by-detection approach was evaluated on a 3.0GHz dual-core dual-CPU, and that their implementation utilized all four cores. On their subset of the CAVIAR dataset with resolution of 384x288

pixels they reported an average throughput of 0.27Hz. 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.

## 6.6 The Effect of Model Uncertainty on the Pedestrian Tracker

Quantitative experiments in Sec. 6.5 demonstrated the effectiveness of our scene representation for tracking pedestrians in the vicinity of relocatable occluders, such as vehicles arriving and departing legal parking spaces. These experiments were conducted in two diverse scenarios that differed in the size of the parking lot, vehicle classes, imaging geometry, and the SNR characteristics of the imaging sensors. In all the test video sequences, alignment of occlusion masks of the instantiated graphical-model layers with parked vehicles in the first frame was approximate, e.g., it did not rely on sub-pixel edge measurements. Subsequent scene updates were performed by our automatic scene-maintenance module that, as shown in Table 6.2, did not perfectly align occlusion masks of the instantiated graphical-model layers with the image truth. Nonetheless, the accuracy of our pedestrian tracker compared favorably to the state of the art.

When our scene representation is applied to a new parking-lot surveillance scenario or a new application domain, there might be multiple possibilities in how to map physical-world objects to classes of relocatable occluders. These choices may influence the generative model for image evidence, its parameter training, the construction of the database of graphical-model layers, and the inference algorithms for instantiating and un-instantiating graphical-model layers on-line. All of these modelling decisions may propagate through the person-tracker that utilizes our scene representation and influence its estimate of the number and the location of persons in the vicinity of relocatable occluders.

A complete analysis of how all of the above choices may interact with each other and propagate into the estimates of the person tracker is a challenge in itself. Instead we treat the combined effects of all of these choices as uncertainty in our scene representation. Later in this section we develop simple models of uncertainty. To accomplish this we turn to the example application of parking-lot surveillance and the corresponding end-to-end system summarized in the diagram in Fig. 3.2. This system comprises an off-line stage of database construction and an on-line stage of scene maintenance and person tracking.

Model uncertainty in the off-line stage is due in part to an approximate mapping between the relocatable objects in the physical world and the designated classes of relocatable occluders, the configuration of activity zones, etc.; it is also due to the uncertainty in estimating parameters of the graphical-model layers in the database. Model uncertainty in the on-line stage is due in part to uncertainty in the location, image-plane orientation, and the occluder class of the instantiated graphical-model layers as well as the topology of the global state space.

As was mentioned in Sec. 4.2.2, uncertainty in our scene model can propagate through the tracking algorithm; in the parking-lot-surveillance example, this uncertainty propagates into the number and location of pedestrians. We can therefore estimate empirically how increasing uncertainty affects the tracking performance measures. To accomplish this, our experiments in this section isolate, to the extent possible, one source of uncertainty at a time and quantify the relation between the amount of uncertainty and the extended performance measures in Table 6.3. These experiments are conducted with the same parameters as in Sec. 6.5.

### 6.6.1 Uncertainty in the Instantiated Scene Model

Our scene representation is designed for uncontrolled, dynamic scenarios. For example, in parking-lot surveillance applications, moving vehicles and pedestrians tend to arrive and depart at random. Depending on the imaging sensor's SNR and its resolution, image frames may exhibit low contrast and low pixel coverage of the relocatable occluders. Furthermore, vehicles and pedestrians may occlude each other in the cameras' non-overlapping fields of view. It is likely that in such scenarios, image-measurement noise and occlusions will propagate through the scene-maintenance algorithm, yielding uncertainty in the scene model.

In theory, the amount of uncertainty in a scene might decrease over time. For example, additional image evidence or feedback from the pedestrian tracker to the scene-maintenance algorithm might be exploited to reduce uncertainty about the scene and update its param-

eters. Since the scene-update module used for the quantitative evaluation in Sec. 6.5 did not update parameters of a layer after it was instantiated, we make the same assumption in our experimental protocol.

In each experiment we restrict uncertainty in the scene to a single parameter in our model. At the time of a layer’s instantiation this parameter is set at random according to the specified amount of uncertainty; all other scene-model parameters are fixed to their image-truth settings for that image frame. The image-truth is determined by a human subject.

We conduct experiments on test video sequences (e) from the PETS2001 dataset and (b) from the COP2007 dataset. For each test video sequence we use the appropriate database of graphical-model layers as defined in Sec. 6.2; all other parameters have the same settings as defined in Sec. 6.5.

**Uncertainty with respect to occluder location.** We represent uncertainty in the 2D image-plane location of an instantiated graphical-model layer as a random variable drawn from a two-dimensional isotropic Gaussian distribution; the distribution’s mean is set to the layer’s image-truth 2D location. There is one such random variable for each instantiated layer, and these random variables are independent. We vary the amount of uncertainty by varying the standard deviation of the Gaussian distributions.

For a test video sequence and its associated image-truth annotation we introduce location uncertainty by replacing an instantiated graphical-model layer’s 2D image location with a sample drawn from our uncertainty model. This process is applied to a graphical-model layer at the time of its instantiation. We then run our pedestrian tracker on the test video sequence with this scene model. Since our uncertainty model is stochastic we repeat the process of sampling from it and running the pedestrian tracker multiple times. We evaluate several uncertainty models by varying the standard deviations of the isotropic Gaussian distributions; in our experiments the standard deviation ranges from  $\sigma = 0.0$  to  $\sigma = 0.15$  times the width in pixels of the layer’s bounding box.

We apply our experimental protocol to the test video sequence (e) from the PETS

2001 dataset and the test video sequence (b) from the COP2007 dataset. We compute the extended performance measures of Table 6.3 as a function of  $\tau_c \in [0.1, 1]$  for each test video sequence and for each uncertainty model, as defined by  $\sigma$ . The average and the standard deviation of these functions over ten samples from the uncertainty model for test sequences (e) and (b) are shown in Fig. 6-6.

Because detailed evaluation of a tracking system with respect to extended performance measures is still uncommon, we briefly summarize typical behavior of these measures as a function of the coverage threshold  $\tau_c$ . A false positive error occurs when a bounding box estimated by the pedestrian tracker does not match any image-truth bounding boxes for a specified  $\tau_c$ . As  $\tau_c$  increases, fewer bounding boxes match the image truth, and the mean false positive  $\overline{\text{FP}}$  tends to increase. A false negative error occurs when an image-truth bounding box does not match any bounding boxes estimated by the pedestrian tracker. In our evaluation protocol in Sec. 6.5 we discount false negatives that do not match any observation regions of our instantiated scene model. Therefore, as  $\tau_c$  increases a greater number of false negatives are ignored so that  $\overline{\text{FN}}$  tends to decrease. A multiple objects error occurs when a bounding box estimated by the pedestrian tracker matches more than one image-truth bounding box. As  $\tau_c$  increases fewer image-truth bounding boxes will match a bounding box estimated by the tracker and therefore  $\overline{\text{MO}}$  typically decreases.

For test sequence (e) all performance measures tend to change gradually with  $\sigma$ . In the case of  $\overline{\text{FP}}$  the averages for all  $\sigma$ 's follows the expected trend: they increase as  $\tau_c$  increases. As  $\sigma$  increases the averages tend to increase gradually: averages corresponding to consecutive  $\sigma$ 's are less than one standard deviation apart. In the case of  $\overline{\text{FN}}$  all the averages follow the expected trend and decrease with  $\tau_c$ . For this performance measure we also observe that as  $\sigma$  varies gradually, the averages gradually shift upward; for consecutive  $\sigma$ 's the averages are in most cases less than one standard deviation apart. The variance increases slightly as location uncertainty tends toward the extreme of our test range at  $\sigma = 0.15$ . In the case of  $\overline{\text{MO}}$  and  $\overline{\text{MT}}$  the averages follow the expected trend and decrease as  $\tau_c$  increases. Varying  $\sigma$  has a less pronounced effect for  $\overline{\text{MO}}$  and  $\overline{\text{MT}}$  than for  $\overline{\text{FP}}$  and  $\overline{\text{FN}}$ .

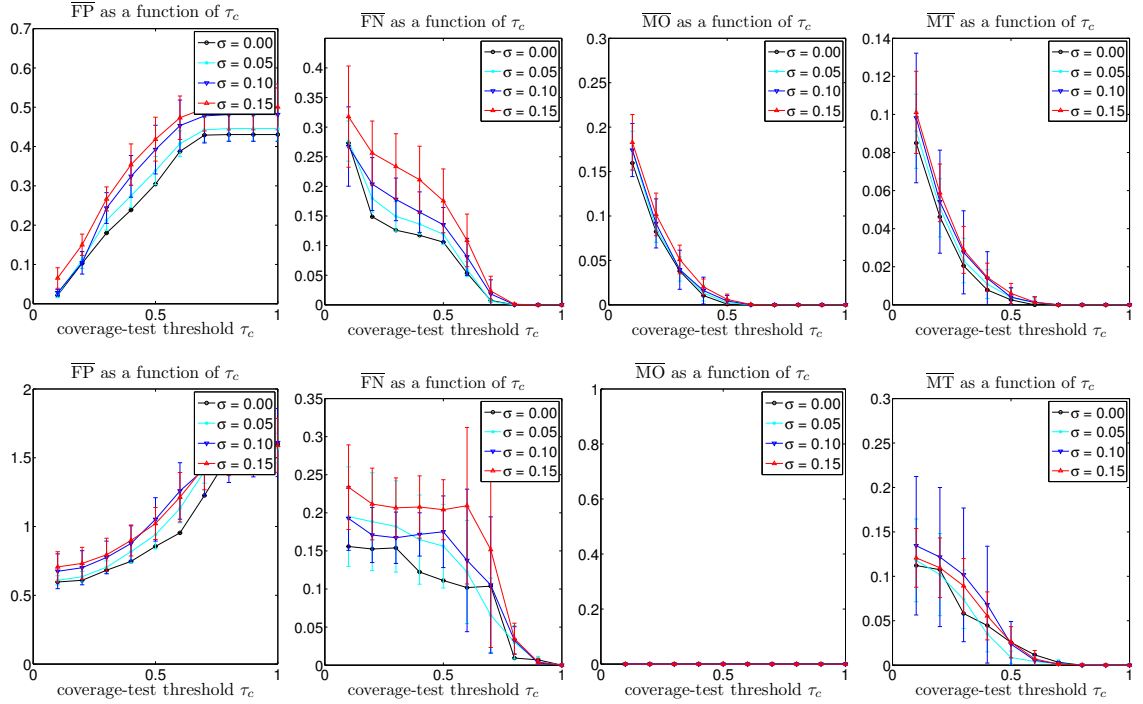
In fact, all the averages are well within one standard deviation of each other regardless of  $\sigma$ .

For test sequence (b) for most of the performance measures there is a gradual change as  $\sigma$  varies gradually. In the case of  $\overline{FP}$  the averages for all  $\sigma$ 's follows the expected trend: they increase as  $\tau_c$  increases. As  $\sigma$  increases the averages tend to increase gradually: averages corresponding to consecutive  $\sigma$ 's are less than one standard deviation apart. As  $\sigma$  tends toward the extreme of its range the average tend to be affected less and all well within one standard deviation of each other. In the case of  $\overline{FN}$  most of the averages follow the expected trend and decrease with  $\tau_c$ , but in several cases the decrease is not strictly monotonic. Nonetheless, in the majority of cases the averages increase gradually with increased  $\sigma$ . Overall the variance tends to increase throughout as  $\sigma$  varies through its range; the average at the extreme of the range of  $\sigma$  is within one standard deviation of the average for the middle of the range for  $\sigma$ . In the case of  $\overline{MO}$  all averages are at zero; this is expected since in test video sequence (b) pedestrians tend to be far apart. In the case of  $\overline{MT}$  the averages follow the expected trend and decrease as  $\tau_c$  increases. The effect of varying  $\sigma$  tends to be less pronounced than in the case of  $\overline{FN}$ . The averages tend to be within one standard deviation of each other. In most cases the variance of  $\overline{MT}$  does not change noticeably with  $\sigma$ .

Common trends in the effects of location uncertainty can be identified among test sequence (e) and test sequence (b). For both test video sequences the averages follow expected behavior as  $\tau_c$  increases. The only exception is the  $\overline{MO}$  performance measure which is identically zero for test sequence (b). For both test video sequences as  $\sigma$  increases the averages tend to increase; this increase tends to be gradual in most cases, but slightly more pronounced in the case of performance measure  $\overline{FN}$  for test sequence (b). Overall  $\overline{MT}$  exhibits less variability with respect to increased location uncertainty. Performance measure  $\overline{FN}$  for test sequence (b) exhibits the most variability with respect to varying  $\sigma$ . Performance measure  $\overline{FN}$  exhibits similar trends as  $\sigma$  increases for both test video sequences.

#### **Uncertainty with respect to the occlusion mask's image-plane orientation.**

We represent uncertainty in the 2D image-plane orientation of the occlusion mask of an



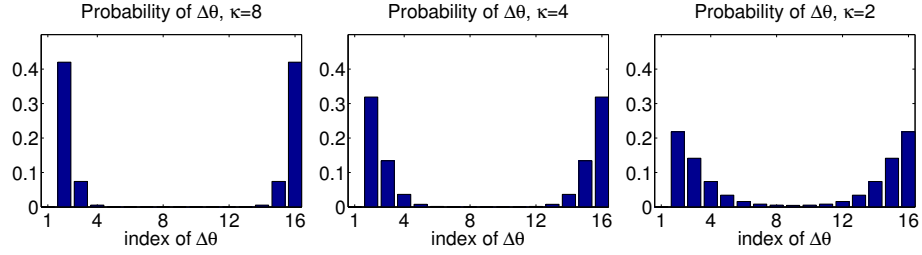
**Figure 6-6:** Location uncertainty of an instantiated graphical-model layer is modelled as a 2D isotropic Gaussian distribution. In our experiments the standard deviation of this Gaussian distribution ranges from  $\sigma = 0.0$  to  $\sigma = 0.15$  times the width in pixels of a layer’s bounding box. For a  $\sigma$  within this range we sample the scene from our uncertainty model and run our pedestrian tracker. This process is repeated ten times. For each performance measure its average and one standard deviation are shown. Top row: test sequence (e); bottom row: test sequence (b).

instantiated graphical-model layer as a random variable drawn from a directional distribution. We choose the von Mises distribution  $p(\theta|\mu, \kappa) \propto \exp\{\kappa \cos(\theta - \mu)\}$  with the concentration parameter  $\kappa$  and the location parameter  $\mu$ . The distribution’s location parameter is set to the occlusion masks’s image-truth orientation. There is one such random variable for each instantiated layer, and these random variables are independent. We vary the amount of uncertainty by varying the concentration parameter  $\kappa$  of the von Mises distribution.

For a test video sequence and its associated image-truth annotation we introduce orientation uncertainty by replacing the orientation of the occlusion mask of an instantiated graphical-model layer with a sample drawn from our uncertainty model. This process is applied to a graphical-model layer at the time of its instantiation. We then run our pedestrian tracker on the test video sequence with this scene model. Since our uncertainty model is stochastic we repeat the process of sampling from it and running the pedestrian tracker multiple times. We evaluate several uncertainty models by varying the concentration parameter of the von Mises distributions; in our experiments  $\kappa \in \{2, 4, 8\}$ .

To simplify implementation, instead of sampling  $\theta \sim p(\cdot | \mu, \kappa)$  we sample  $\Delta\theta \sim p(\cdot | 0, \kappa)$  and add such  $\Delta\theta$  to the image-truth  $\mu$  with circular wrap-around. Since our database of graphical-model layers is indexed by discrete  $\theta$ ’s, the distribution from which we sample  $\Delta\theta$  is also discretized. To ensure that samples drawn from our uncertainty model always change an occlusion masks’s orientation we set the probability mass for  $\Delta\theta = 0$  to zero. Examples of such discretized distributions for  $\Delta\theta$  with  $\kappa \in \{2, 4, 8\}$  are shown in Fig. 6-7.

We apply our experimental protocol to the test video sequence (e) from the PETS 2001 dataset and the test video sequence (b) from the COP2007 dataset. We compute the extended performance measures of Table 6.3 as a function of  $\tau_c \in [0.1, 1]$  for each test video sequence and for each uncertainty model, as defined by  $\kappa$ . The average and the standard deviation of these functions over ten samples from the uncertainty model and for test sequences (e) and (b) are shown in Fig. 6-8.



**Figure 6.7:** Distribution of  $\Delta\theta$  with location  $\mu = 0$  and the concentration parameter  $\kappa \in \{2, 4, 8\}$ . To ensure that a non-zero  $\Delta\theta$  is drawn, the probability mass for the first bin is set to zero and the distribution is re-normalized.

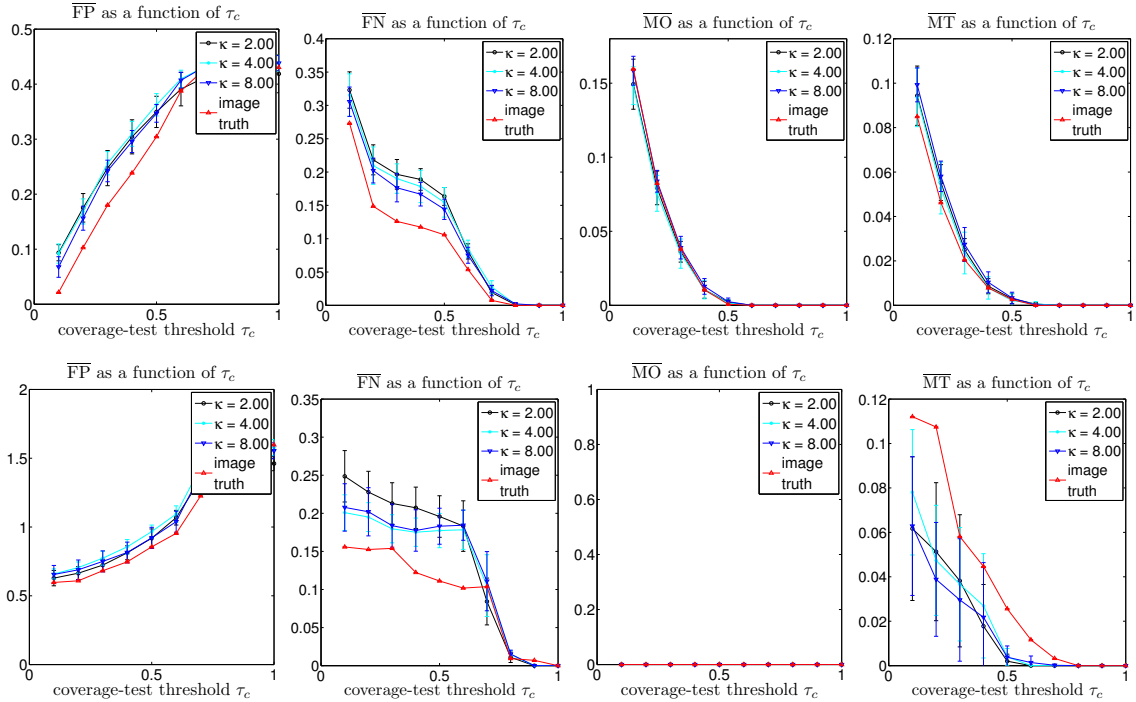
We state again the expected behavior of the extended performance measures as a function of the coverage threshold  $\tau_c$ . A false positive error occurs when a bounding box estimated by the pedestrian tracker does not match any image-truth bounding boxes for a specified  $\tau_c$ . As  $\tau_c$  increases we expect  $\overline{\text{FP}}$  to increase. A false negative error occurs when an image-truth bounding box does not match any bounding boxes estimated by the pedestrian tracker. In accordance with our evaluation protocol in Sec. 6.5, as  $\tau_c$  increases a greater number of false negatives are ignored and therefore  $\overline{\text{FN}}$  tends to decrease. A multiple-objects error occurs when a bounding box estimated by the pedestrian tracker matches more than one image-truth bounding box and tends to decrease as  $\tau_c$  increase; similar trend is typically followed by  $\overline{\text{MO}}$ .

For test sequence (e) all performance measures tend to change gradually with  $\kappa$ . In the case of  $\overline{\text{FP}}$  the averages for all  $\kappa$ 's follows the expected trend: they increase as  $\tau_c$  increases. When uncertainty at the lower end of the range,  $\kappa = 8.0$ , is introduced,  $\overline{\text{FP}}$  increases by about one-and-a-half standard deviations. Further increase in uncertainty, accomplished by lowering  $\kappa$  toward  $\kappa = 2.0$  does not seem to have a pronounced effect: averages corresponding to all  $\kappa$ 's are all within one standard deviation of each other. In the case of  $\overline{\text{FN}}$  all the averages follow the expected trend and decrease with  $\tau_c$ . For this performance measure we also observe that as uncertainty of  $\kappa = 8.0$  is introduced, the average increases by about two standard deviations. Further increase in uncertainty, accomplished by lowering  $\kappa$ , leads to a gradual and monotonic increase in  $\overline{\text{FN}}$ . The variance

remains roughly the same for all  $\kappa$ 's. In the case of  $\overline{\text{MO}}$  and  $\overline{\text{MT}}$  the averages follow the expected trend and decrease as  $\tau_c$  increases. Varying  $\kappa$  tends to have less effect on  $\overline{\text{MO}}$  than on either  $\overline{\text{FP}}$  or  $\overline{\text{FN}}$ . The variance appears unaffected by  $\kappa$  and remains the same for each  $\tau_c$ . Averages for  $\overline{\text{MT}}$  also exhibit little affect of changing  $\kappa$ .

For test sequence (b) for half of the performance measures there is gradual change with  $\kappa$ . In the case of  $\overline{\text{FP}}$  the averages for all  $\kappa$ 's follows the expected trend: they increase as  $\tau_c$  increases. As  $\kappa$  increases the averages tend to increase somewhat and they all remain within one standard deviation from each other and from  $\overline{\text{FP}}$  obtained with image truth. In the case of  $\overline{\text{FN}}$  most of the averages follow the expected trend and decrease with  $\tau_c$ , but in several cases the decrease is not always monotonic. When uncertainty of  $\kappa = 8.0$  is introduced the average  $\overline{\text{FN}}$  increases by about one-and-a-half standard deviations. Further increase in uncertainty of  $\kappa = 4.0$  has no noticeable effect on  $\overline{\text{FN}}$ , and when  $\kappa = 2.0$  the average increases by about one standard deviation. This behavior may be attributed in part to the artifacts of sampling from our discrete uncertainty model. As  $\tau_c$ 's increases from 0.5 to 1.0 the effects of discrete sampling become less pronounced, and all the averages begin to converge together. For all  $\kappa$ 's the averages remain within one standard deviation from each. In the case of  $\overline{\text{MO}}$  all averages are at zero; this is expected since in test video sequence (b) pedestrians tend to be far apart. In the case of  $\overline{\text{MT}}$  the averages follow the expected trend and decrease as  $\tau_c$  increases. The variance tends to be larger than for  $\overline{\text{FP}}$  and  $\overline{\text{FN}}$ . The averages for all  $\kappa$ 's appear somewhat below  $\overline{\text{MT}}$  corresponding to the image truth. However, these averages are within one standard deviation from each other, and for some  $\tau_c$ 's they are within one standard deviation of  $\overline{\text{MT}}$  for the ground truth.

Common trends in the effects of occlusion masks's orientation uncertainty can be identified among test sequence (e) and test sequence (b). For both test video sequences the averages follow expected behavior as  $\tau_c$  increases. The only exception is the  $\overline{\text{MO}}$  performance measure which is identically zero for test sequence (b). For both test video sequences as  $\kappa$  increases the averages for  $\overline{\text{FP}}$  and  $\overline{\text{FN}}$  tend to increase; this increase tends to be gradual in most cases, but slightly more pronounced in the case of performance measure  $\overline{\text{FN}}$  for



**Figure 6-8:** Orientation uncertainty of the occlusion mask of an instantiated graphical-model layer is modelled as a von Mises distribution with concentration  $\kappa$ ; in our experiments  $\kappa \in \{2.0, 4.0, 8.0\}$ . For a  $\kappa$  in this set we sample the scene from our uncertainty model and run our pedestrian tracker; a total of ten scenes are sampled. For each performance measure its average and one standard deviation are shown. Top row: test sequence (e); bottom row: test sequence (b).

test sequence (b). Overall  $\overline{MO}$  exhibits less variability with respect to increased orientation uncertainty. Performance measure  $\overline{MT}$  for test sequence (b) exhibits the most variability with respect to varying  $\kappa$ .

**Uncertainty with respect to an occluder’s class.** We represent uncertainty in the occluder class of an instantiated graphical-model-layer as a random variable drawn from a multinomial distribution. There is one such random variable for each instantiated layer, and these random variables are independent. We increase the amount of uncertainty by increasing the entropy of the multinomial distribution.

For a test video sequence and its associated image-truth annotation we introduce uncertainty with respect to occluder’s class by replacing an instantiated graphical-model

layer’s class with a sample drawn from our uncertainty model. This process is applied to a graphical-model layer at the time of its instantiation. We then run our pedestrian tracker on the test video sequence with this scene model. Since our uncertainty model is stochastic we repeat the process of sampling from it and running the pedestrian tracker multiple times. We evaluate several uncertainty models by varying the entropy of the multinomial distribution.

To simplify the implementation we do not directly sample a multinomial distribution, but employ hierarchical sampling to specify the occluder class of a graphical-model layer. A Bernoulli random variable is sampled, and if the sample equals zero, the occluder’s class is specified according to the image truth. If the sample does not equal zero, the occluder’s class is sampled from a uniform distribution over all occluder classes except the one specified by the image truth; the probability of drawing the same occluder class as image truth is set to zero.

We apply our experimental protocol to the test video sequence (e) from the PETS 2001 dataset and the test video sequence (b) from the COP2007 dataset. We compute the extended performance measures of Table 6.3 as a function of the Bernoulli distribution’s parameter  $p \in [0.0, 1.0]$  for each test video sequence and for each uncertainty model, as defined by  $p$ . The average and the standard deviation of these functions over ten samples from the uncertainty model for test sequences (e) and (b) are shown in Fig. 6.9.

Because analyzing the effect of  $\tau_c$  on the extended performance measures is still uncommon we briefly summarize the expected trends; a complete explanation is provided earlier in this section. As  $\tau_c$  increases we would expect  $\overline{\text{FP}}$  to increase. The remaining performance measures,  $\overline{\text{FN}}$ ,  $\overline{\text{MT}}$ , and  $\overline{\text{MO}}$  typically decrease as  $\tau_c$  increases.

For test sequence (e) all performance measures tend to change gradually with  $p$ . In the case of  $\overline{\text{FP}}$  the averages for all  $p$ ’s follows the expected trend: they increase as  $\tau_c$  increases. As  $p$  increases from  $p = 0.0$  to  $p = 0.25$  the average  $\overline{\text{FP}}$  increases; this increase is within one standard deviation. Further increases in  $p$  yield gradual and for the most part monotonic increases in the averages. The averages for consecutive  $p$ ’s are less than

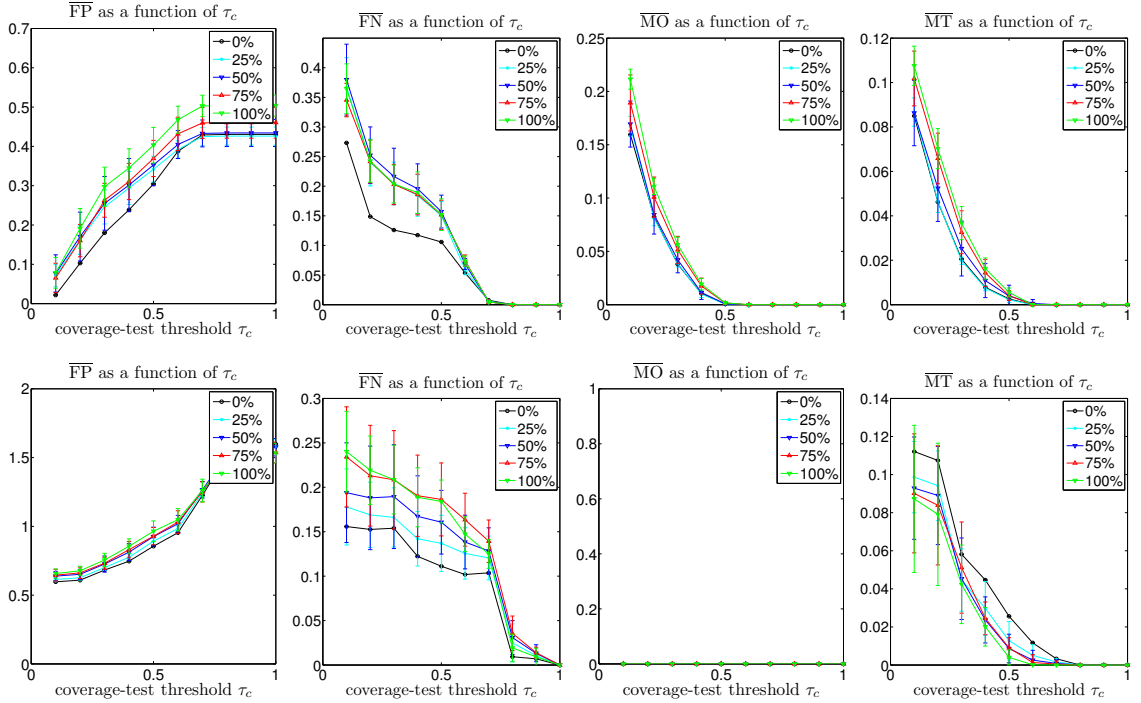
one standard deviation apart. The variance remains roughly the same for all  $p \geq 0.25$ . In the case of  $\overline{\text{FN}}$  all the averages follow the expected trend and decrease with  $\tau_c$ . As  $p$  changes from  $p = 0.0$  to  $p = 0.25$  the average increases by roughly two standard deviations. Further increase in  $p$  does not have a noticeable effect on the averages; they remain close the average for  $p = 0.25$  and well within one standard deviation from each other. The variance also remains the same for all  $p \geq 0.25$ . In the case of  $\overline{\text{MO}}$  the averages follow the expected trend by decreasing as  $\tau_c$  increases. The variance tends to be the same for all  $p$ 's and for all  $\tau_c$ 's. All of the averages are well within one standard deviation of each other. In the case of  $\overline{\text{MT}}$  a trend similar to that for  $\overline{\text{MO}}$  is observed. The averages for  $p = 0.0$  and  $p = 1.0$  are more noticeably different than for  $\overline{\text{MO}}$ , but all averages are well within one-and-a-half standard deviations from each other. The variance does not change noticeably as  $p$  and  $\tau_c$  vary.

For test sequence (b), half of the performance measures exhibit a gradual change with changing  $p$ . In the case of  $\overline{\text{FP}}$  the averages for all  $p$ 's follow the expected trend: they increase as  $\tau_c$  increases. As  $p$  increases the averages tend to increase gradually and monotonically. All the averages remain within one standard deviation from each other and from  $\overline{\text{FP}}$  that corresponds to the image truth. Overall the variance remains roughly the same for all  $p$ 's and  $\tau_c$ 's. In the case of  $\overline{\text{FN}}$  all of the averages follow the expected trend and decrease with  $\tau_c$ , but the decrease is not always strictly monotonic. As uncertainty is introduced at  $p = 0.25$ , the average uniformly shifts up by less than one standard deviation. Further increasing  $p$  has the effect of shifting the averages up somewhat and within one standard deviation of each other. The variance in the case of  $\overline{\text{FN}}$  is somewhat larger than for  $\overline{\text{FP}}$ ; it remains the same for all  $p$ 's and decreases somewhat as  $\tau_c$  increases. In the case of  $\overline{\text{MO}}$  all averages are at zero; this is expected since in test video sequence (b) pedestrians tend to be far apart. In the case of  $\overline{\text{MT}}$  the averages follow the expected trend and decrease as  $\tau_c$  increases; the averages for  $p > 0$  are slightly below  $\overline{\text{MT}}$  for the image truth. The variances are roughly equal to the variances of  $\overline{\text{FN}}$  and decrease somewhat as  $\tau_c$  increases. Overall the averages stay within one standard deviation of each other and  $\overline{\text{MT}}$  for the image truth.

Common trends in the effects of the uncertainty in an occluder’s class can be identified among test sequence (e) and test sequence (b). For both test video sequences as  $\tau_c$  increases the averages follow the predicted trends. The only exception is the  $\overline{\text{MO}}$  performance measure which is identically zero for test sequence (b). For both test video sequences as  $p$  increases the averages for  $\overline{\text{FP}}$  and  $\overline{\text{FN}}$  tend to increase; this increase tends to be gradual in all cases, but the difference between  $p = 0.0$  and  $p = 0.25$  is somewhat more pronounced in the case of performance measure  $\overline{\text{FN}}$  for test sequence (e). Overall  $\overline{\text{MO}}$  for test sequence (e) and  $\overline{\text{FP}}$  for test sequence (b) are the least affected by increased uncertainty in occluder class. Since our uncertainty model is discrete there is no a priori expectation that gradual changes in  $p$  would yield gradual changes in the averages; it is therefore somewhat surprising to see monotonic and gradual changes in  $\overline{\text{FN}}$  for test sequence (b). For both test sequences the variances tended to be relatively unaffected by either  $p$  or  $\tau_c$ .

**Summary of the scene-uncertainty experiments.** In our experiments the two test video sequences are chosen to explore diverse scenarios. The differences are due in part to the imaging sensors, the viewing geometry, the apparent size of pedestrians, and the number of relocatable occluders; consequently a separate database of graphical-model-layers is constructed for each scenario. In spite of these differences the effects of scene uncertainty on the pedestrian tracker share commonalities across these two scenarios.

Overall, a gradual increase in the amount of uncertainty yields a gradual change in each performance measure. In some cases—e.g., the effect of orientation uncertainty on  $\overline{\text{FP}}$ —a performance measure is affected as the amount of uncertainty is increased up to a point, but further increase in the amount of uncertainty has no appreciable effect. Overall the variance of a performance measure with respect to random samples from our uncertainty model remained roughly the same across the two sequences. The effect of uncertainty is somewhat more pronounced on performance measure  $\overline{\text{FN}}$ . This may be due in part to the learned parameters for binary image evidence in our generative model and the parameters of our pedestrian tracker. The effect of uncertainty tends to be least pronounced on  $\overline{\text{MO}}$  and  $\overline{\text{MT}}$ , due in part to the penalty on overlapping tracks in our pedestrian tracker.



**Figure 6-9:** Occluder-class uncertainty of an instantiated graphical-model layer follows a hierarchical model. A draw from the Bernoulli distribution with parameter  $p$  determines if an occluder’s class follows the image truth; otherwise the occluder’s class is drawn randomly from a uniform distribution over the occluder classes other than the one specified by the image truth. In our experiments  $p \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ ; in this figure  $p$  is expressed as percentages. For a  $p$  in this range we sample the scene from our uncertainty model and run our pedestrian tracker. This process is repeated ten times. For each performance measure its average and one standard deviation are shown. Top row: test sequence (e); bottom row: test sequence (b).

## Chapter 7

# A Guide to Applying Layered Graphical Models to New Problem Domains

In Sec. 6.5 of the preceding chapter we demonstrated the effectiveness of our scene representation for parking-lot surveillance applications. Following that, in Sec. 6.6 we focused on quantifying the accuracy of our person tracker in new scenarios and problem domains. Applying our formulation in novel settings would necessitate making numerous design choices, and modelling the effects of all such choices on our scene representation would not have been tractable. Therefore, we chose to view the combined effects of all the design choices as uncertainty in our model and developed tractable representation for uncertainty in our scene representation. Given the uncertainty models for scenes comprising the instantiated graphical-model layers, we were able to quantify how uncertainty propagated into the output of our pedestrian tracker.

In this chapter we present a guide for applying our general formulation to new scenarios and problem domains. Our goals include enumerating the relevant design choices and presenting some guidelines for choosing among them. Such design choices and guidelines are grouped according to our formulation in Chapter 3. In addition, all of the design guidelines are unified by a common theme of realizing the full benefit of our scene representation for person tracking. Achieving such a goal requires taking into consideration the computational complexity of the person tracker and the propagation of uncertainty into the tracker's output.

Our formulation of modelling dynamic scenes with graphical-model layers imposes few

constraints on the overall design of a complete system. In particular, an algorithm designer is free to specify the inference algorithm for scene maintenance. Among such scene-maintenance algorithms there may be a difference in the extent to which feedback from other modules, e.g., a person tracker, is used to further reduce uncertainty in the instantiated graphical-model layers. However, to make the discussion concrete, we motivate this chapter by considering a complete system without complex interdependencies; the system developed for our parking-lot surveillance application falls into such category, because the person tracker and the scene-maintenance algorithm did not share information.

## 7.1 Designing Graphical-Model Layers

We can think of designing graphical-model layers for a new scenario or a problem domain as establishing a mapping between the physical world and our representation defined in terms of the activity zones, the occlusion mask, and the observation regions; this representation was summarized in the previous chapters by a probabilistic graphical model.

**Specifying classes of the relocatable and static occluders.** In the case of parking-lot surveillance, we identified as relocatable occluders passenger vehicles that arrived in and left legal parking spaces. To account for variation in vehicles' shapes we defined a hierarchy of occluders based on the most common vehicle types. In our experiments the 3D vehicle volume was set slightly lower for the PETS2001 dataset as European vehicles tend to be smaller than their US counterparts.

In a new scenario the variability in the shape of physical-world relocatable objects and the motion patterns of people around them may also be effectively represented by a hierarchy of the relocatable-occluder classes. In principle, a deeper hierarchy might yield a more accurate scene representation in terms of the overlap between a graphical-model layer's occlusion mask and projection of the 3D scene in a camera's field of view; such a hierarchy might also offer more refined modelling of the motion patterns of people near the relocatable occluders. In practice, maintaining a scene representation with a deep hierarchy of the relocatable occluder classes might be too computationally expensive. Fortunately,

our experiments in Sec. 6.6 indicate that gradual increase in scene uncertainty tends to propagate gradually into the tracker’s output. Therefore, depending on the application requirements, one could choose to reduce the computational complexity of the complete system at the expense of some uncertainty in the scene representation; the available image evidence may also influence the design of the relocatable-occluder class hierarchy.

Our scene representation extends to scenarios that include a mixture of relocatable and static occluders. Indeed, the distinction between these two types of occluders is likely to be immaterial to a person tracker, and from the standpoint of a scene-update algorithm a static occluder is a relocatable occluder whose lifespan has been fixed a priori. In principle, if the location of the static occluders is known a priori, introducing a more extensive hierarchy of the static-occluder classes may not increase the computational complexity of the complete system. In practice, and similar to the case of the relocatable occluders, the usefulness of any such hierarchy is likely to depend on the available image evidence.

**Defining the topology of activity zones and the transitions between them.**

For our example application of parking-lot surveillance, we applied our scene representation to modelling the physical processes in the ground plane. Therefore, our activity zones mapped to regions in the ground plane in the vicinity of a relocatable occluder. In our implementation, square image-plane regions formed a ring around their relocatable occluder; these squares did not overlap each other or the relocatable occluder except at their boundaries. The transition probabilities between activity zones were specified as uniform to their immediate neighbors; jumping over the relocatable occluder was disallowed.

In a new scenario or a problem domain, activity zones may be defined to realize the full benefit of our representation. In particular, activity zones can be mapped to non-coplanar regions to better capture the process in the physical 3D world. For some applications the computational complexity of a person tracker might be of lesser concern than uncertainty in the tracker’s estimates of the number and location of the persons. This trade-off can be achieved in part by increasing the density of the activity zones and by allowing their overlap; such a strategy was successfully explored by [Fleuret et al., 2008]. Allowed transitions

between the activity zones can be defined in a way that captures a person’s likely motion patterns. For example, the relocatable object which owns these activity zones is likely to impose physical constraints on where a person can move.

**Defining the observation regions and the generative model for image evidence.** In the case of parking-lot surveillance we established a one-to-one correspondence between the ground-plane activity zones owned by a graphical-model layer and its image-plane observation regions. An observation region comprised an image-plane rectangle and variable indicating the depth-order between this rectangle and the layer’s occlusion mask. The projection of a person onto the image plane was modelled as a binary rectangular mask whose extent coincided with an observation region. Our generative model for the image evidence was motivated by the poor resolution and low contrast of the video sequences acquired in the surveillance applications. Therefore we assumed that the image evidence took the form of binary moving-pixel indicators estimated by an off-the-shelf background subtraction algorithm.

In a new scenario or problem domain, observation regions may be defined to exploit the projected motion patterns of persons in the corresponding activity zones. Indeed, the shape of the observation regions may vary among the activity zones owned by the same graphical-model layer and across the classes of relocatable occluders. Depending on the viewing geometry and the imaging sensor resolution, the projection of a person onto the image plane might be more accurately captured by imposing a structure within the observation regions. In some cases such a structure may be inspired by a representation initially proposed for free-space tracking. For example, the image-plane-based approach of [Nejhum et al., 2010] for person tracking models a target as set of *articulated blocks* whose spatial arrangement and size vary with time. In adapting this approach to our scene representation, one could consider partitioning a rectangular observation region into blocks so that a person’s image-plane projection maps to a subset of such blocks.

The design of the generative model of image evidence in the observation regions may take into account how uncertainty in such a model would propagate into the person-

tracker’s output. As was demonstrated in Sec. 6.5 our person tracker achieves state-of-the-art performance with binary, noisy features. In scenarios with high image resolution and high signal-to-noise ratio, a generative model that captures the projection of a person inside an observation region may enable a more accurate estimate of the number and the location of the persons in our scene representation. If the available image evidence is sufficient to acquire person-specific appearance models, it may be possible to disambiguate the depth-order of the locations of persons whose projections overlap in the image plane. In some cases, the same algorithms used for free-space inference for the location and the depth-order of people may be applicable to observation regions. For example, if color information is available, representing the image projection of an upright person as a collection of horizontal slices and modelling the appearance of each slice non-parametrically might be appropriate; such a representation, proposed by [Elgammal and Davis, 2001] and was applied to tracking closely-spaced people.

### 7.1.1 Specifying Parameters

Our scene representation offers several advantages with regard to parameter learning. First, our graphical-model layers are pre-computed into a database during the off-line stage, lowering the computational complexity of the on-line stage; for real-time applications the ability to transfer computationally-expensive parameter learning, e.g., occlusion-mask learning, to the off-line stage is desirable. Second, as our example implementation for parking-lot surveillance demonstrates, parameter-learning can be stratified so that at each stage only a subset of parameters needs to be estimated. Third, because our representation is occluder-centric, some of the parameters are viewpoint independent and may therefore be learned under advantageous imaging conditions.

**Scene geometry.** In our implementation the scene geometry was modelled by a projection matrix estimated from the relations between scene features of typical office-building environments, e.g., lighting posts and pedestrians in a parking lot; these relations were derived from the knowledge of typical parking-lot layouts rather than cite-specific

building maps. In other scenarios the projection matrix can be recovered from an image sequence of a walking person, e.g., [Lv et al., 2006]. Because our person tracker does not depend on an explicit 3D-to-2D mapping, there is no requirement that the database of graphical-model layers be constructed with such a mapping. Therefore, in some scenarios it may be practical to learn the scene geometry that is specific to the targets of interest. For example, in the work by [Renno et al., 2002], a scene geometry was learned to estimate the expected bounding-box extent of pedestrians and vehicles as a function of image-plane coordinates.

**Occlusion masks of the relocatable objects.** In our implementation an image-plane occlusion mask of a relocatable object is represented as a set of binary random variables, one per pixel. These occlusion masks are acquired by applying standard computer-rendering techniques to our polygonal vehicle models. A vehicle model belongs to one of several allowed vehicle classes and is relatively coarse; it does not take into account variations due to different vehicle manufacturers. In some scenarios it may be practical to acquire the occlusion masks without an explicit rendering stage. For example in the work by [Titsias and Williams, 2006], the objects’ occlusion masks are automatically learned and clustered according to views via an EM algorithm. Although the computational complexity of this algorithm—as noted in Sec. 6.4—may not be a good match for real-time requirements, its benefits may be fully realized during our off-line database construction stage.

Occlusion masks of the static occluders in the scene may be acquired using the same procedure as for the relocatable occluders, e.g., via computer-graphics rendering of polygonal models. Depending on the application domain it may be practical to learn the occlusion masks of static objects in the scene without explicit rendering of 3D models. For example, in the work by [Renno et al., 2007], depth-ordered occlusion masks of multiple subway turnstiles were learned by accumulating image evidence in the vicinity of person tracks over time.

**Activity-zone transition probabilities.** In our experiments the transition proba-

bilities between activity zones owned by the same graphical-model layer were defined so as to make the self-transition and the transition to immediate neighbors equally likely. A similar approach to setting transition probabilities uniformly for a subset of zones chosen based on a person’s expected mobility constraints was taken by [Fleuret et al., 2008]. In some application domains it may be practical to learn zone transition probabilities from trajectories of people in the scene as was done in, e.g., as was proposed by [Demirdjian et al., 2002], or possibly directly from the image evidence. Since transition probabilities are specified in the occluder-centric representation and do not depend on an instantiated scene, the learning algorithm and the training data may be chosen to minimize the uncertainty in the model.

Transition probabilities between the activity zones owned by different instantiated graphical-model layers were set uniformly among the subset of such zones that passed our connectivity test in Sec. 3.1. In some scenarios, transition probabilities between the activity zones whose observation regions are proximate in the image plane may take into account their relative sizes and overlap; depending on the representation of the scene geometry, ground-plane mobility constraints may also be taken into account. If it is practical to acquire labelled training data comprising the instantiated graphical-model layers and image-plane trajectories of people in the scene, it may be possible to learn transitions between the activity zones owned by different graphical-model layers in a supervised manner.

## 7.2 Designing a Scene-Maintenance Algorithm

The scene-maintenance algorithm may be chosen to minimize the amount of uncertainty propagating into the person tracker’s output. In our experiments this was accomplished by using image features capable of discriminating when an occlusion mask of a vehicle has come to rest or started moving. Depending on a scenario or application domain, it might be possible to reduce the uncertainty in an instantiated scene after a graphical-model layer has been instantiated. For example, in the case of parking-lot surveillance, vehicles tend to occupy legal parking spaces longer relative to the typical sensor’s frame

rate. Therefore, it might be possible to continue accumulating image evidence relevant to the layers' parameters, e.g., 2D location, after their instantiation. In the case of parking-lot surveillance, uncertainty may be also reduced by incorporating a feedback from the person tracker, e.g., by running the person tracker backward in time as was done by [Ryoo et al., 2010].

### 7.2.1 Specifying Parameters

Our scene representation offers several advantages for learning the parameters of a scene-maintenance algorithm.

**The dynamics of the relocatable occluders.** The benefit of our implicit-3D scene representation can be realized to reduce the uncertainty in the estimated dynamics model. For example, the dynamics of the relocatable objects may be initially learned in a computationally-convenient coordinate system, e.g., the ground plane, and using informative image features. Such a dynamics model might then be transferred to the viewpoint anticipated in the scenario of interest. In our example application of parking-lot surveillance, a vehicle's dynamics was mapped into a distribution over the parameters of an occlusion mask in the database of graphical-model layers.

**Interaction patterns of the relocatable occluders and people in their vicinity.** In our example parking-lot surveillance application, joint dynamics of pedestrians and vehicles could be established based on the physical constraints. Indeed, it was assumed that a pedestrian could not emerge from or enter a moving vehicle and that a vehicle would typically not drive over a pedestrian. In a new application domain, it might be possible to acquire a joint model for the motion patterns of relocatable objects and people in their vicinity. In some cases the algorithms for inferring interactions between people may be applicable to inferring interactions between people and relocatable objects. For example, if labelled training data comprises bounding boxes for targets in the scene, the coupled-HMM approach of [Oliver et al., 2000] can be applied to learn pairwise interaction patterns.

### 7.3 Designing a Person Tracker

The person tracker may be chosen to minimize the uncertainty of the estimated location and the counts of people in a scene. In some application domains, e.g., real-time surveillance, the design of person tracker may be influenced by the computational-complexity requirements. As demonstrated in Chapter 6 for the parking-lot surveillance application, our scene representation allows us to design a person tracker whose computational complexity and accuracy compare favorably with the state of the art.

**Providing the estimates of uncertainty in person locations.** Depending on the application domain, the person tracker designed for our scene representation may be a part of a larger system. In such cases, uncertainty in the person-tracker’s output may propagate into the rest of the system. Therefore, it may be desirable for the person tracker to yield an estimate of the uncertainty in the location of people in the scene. In our RJ MCMC tracker adapted to the instantiated graphical-model layers, the estimated locations of people in the scene were represented as samples in the Markov chain; the density of such samples in relation to the activity zones could be taken as a measure of uncertainty in the tracker’s output. In our Viterbi-based tracker, the estimated locations of people in the scene were represented as non-overlapping paths in the Viterbi trellis; marginal distributions—*beliefs*—over activity zones at each time step could be obtained using the sum-product algorithm [Kschischang et al., 2001, Bishop, 2006] and be taken as a measure of certainty in the person tracker’s output.

**Handling the overlap of people in the observation regions.** In some scenarios or application domains, it might be advantageous to allow multi-person occupancy in the activity zones. In the observation regions of such activity zones the projections of multiple people can overlap. The uncertainty in the depth-order of such projections might propagate into the tracker’s estimates of the number and the location of people in the vicinity of the relocatable occluders.

The benefit of handling the overlap of people in the observation regions also extends

to the graphical-model layers designed with a single-occupancy constraint on the activity zones. For example, two persons may occupy distinct activity zones whose corresponding observation regions overlap, but the depth-order of these observation regions, and hence the persons, is not specified by the scene model; this might happen if the activity zones are owned by different graphical-model layers.

In the application of our scene representation to parking-lot surveillance, pedestrians were modelled as binary masks. Such binary masks did not convey information about the depth-order. Therefore, our Viterbi-based pedestrian tracker realized the benefit of depth-ordered targets only for the observation regions that had such a depth-order specified in our instantiated scene.

In some scenarios the generative model for the image evidence in the observation regions might convey information about the depth-order of people in the scene. It may then be possible to apply to algorithms that were proposed for inferring the depth-order of people in free-space to estimating the depth-order of people in the observation regions. One example would be a scenario where color information is available—in this case, the non-parametric person’s appearance representation and the *occlusion-hypothesis* evaluation algorithm of [Elgammal and Davis, 2001] may be applicable. Another example would be a scenario where it is practical to learn a family of body-part detectors, e.g., for the upper body, the torso, and the legs—in this case the distributed-data-association approach of [Yu et al., 2008] may be applicable. If the inference over the number and the location of people in the scene is approximated via RJ MCMC, then such an inference algorithm might be extended as a Markov sequential object process [van Lieshout, 2008] to estimate the number, the location, and the depth-order of people.

### 7.3.1 Specifying Parameters

Parameters of a person tracker may be specified to achieve the required accuracy of the estimated number and the locations of people in the vicinity of the relocatable occluders. This general principle was applied to our two examples of pedestrian trackers for the

parking-lot surveillance application. In the first example, parameters of the RJ MCMC person tracker in Sec. 4.1 included the probabilities of the *birth* and *death* moves; these probabilities were specified according to the expected rate of the arrival and departure of pedestrians in the parking lot. In the second example, parameters of the Viterbi-based tracker in Sec. 4.2 included multiplicative factors of the activity-zone likelihood function; these parameters were specified to increase the likelihood of a valid track in the presence of noisy image evidence in video frames.

Our implicit-3D scene representation comprising instantiated graphical-model layers can lead to straightforward algorithms for learning the person-tracker parameters. Among such learning algorithms we highlight two non-exclusive possibilities: semi-supervised training with 2D image labels, and supervised training with examples obtained from our generative model.

In the semi-supervised learning formulation, image-plane annotation—e.g., bounding rectangles of persons in the scene—are made available to the training algorithm as examples of correct tracks; examples of incorrect tracks may be acquired from the bounding boxes mis-aligned with respect to the image truth. Some parameter-training algorithms may require that the training sequences be specified in terms of the activity zones rather than image-plane annotations; this was the case for the training algorithm for our Viterbi-based tracker in Sec. 4.2. In such cases, the activity zones may be regarded as missing labels. These missing labels would then be estimated jointly with the parameters of the tracking algorithm. If the training algorithm is formulated as a likelihood maximization with respect to the positive and negative training examples, the missing labels for activity zones may be inferred in a standard fashion using the EM algorithm.

An example of a fully-supervised parameter-learning formulation is one where the positive and negative training sequences are defined on the activity zones. In this case, learning the tracker parameters may reduce to an optimization problem, such as linear programming in Sec. 4.2. However, in some scenarios acquiring labelled training sequences with the help of human subjects might not be practical. Instead we may choose to generate

the sequences of activity zones and the corresponding image evidence by sampling from a scene comprising the instantiated graphical-model layers; this approach was taken in our experiments in Sec 6.5 and Sec. 6.6. In principle, semi-labelled training sequences may be combined with the sequences obtained from our generative model, thus increasing the overall training-set size and possibly reducing the uncertainty in the estimated parameters.

## Chapter 8

# Discussion and Conclusions

In the final chapter of the thesis we summarize the main contributions and open issues in the work that we have described. We also point out potential directions for future work.

### 8.1 Main Contributions

This section provides a summary of contributions made in this thesis: A layers-of-graphical-models formulation that enables us to track partially-occluded persons interacting with relocatable occluders.

### 8.2 Future Work

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 [White and Shah, 2007]. 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. Our tracking algorithm in Sec. 4.2 may be improved by optimizing over all pedestrian tracks jointly as implemented in [Ma et al., 2009] based on [Wolf et al., 1989, Castañon, 1990].

As future work we aim to develop a unified formulation for tracking on activity zones and in free-space. In order to define a first-order Markov process for a person’s motion through the entire scene a state space needs to be defined first. This state space must enable switching between discrete activity-zones and continuous free-space location. During tracking, image evidence in the entire video frame would be used to infer the number and locations of persons in the entire scene. The free-space and the activity-zone trackers could be combined as proposed in e.g., [Leichter et al., 2006, Du and Piater, 2008]. We look forward to developing such a formulation and validating it with a free-space tracker that is well-matched for our challenging datasets; examples of potentially-applicable approaches include [Breitenstein et al., 2011, Dunne and Matuszewski, 2011].

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 [Kang et al., 2003, Vacchetti et al., 2004] and fuse the image evidence across all the views [Khan and Shah, 2009]. A multi-view layered representation has been proposed in [Reid and Connor, 2010] but experimental validation was performed with small camera displacements. Another approach is to maintain one global 3D representation [Vedula et al., 1998, Khan and Shah, 2009] 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.

Future work on extending our scene-maintenance subsystem would first focus on incorporating a free-space vehicle tracker. Such a tracker, e.g., [Pece, 2006, Ottlik and Nagel,

2007, Atev and Papanikolopoulos, 2008, Leotta and Mundy, 2011], could yield an improved predictor of a vehicle's arrival. Furthermore, one would expect that by integrating observations of a vehicle over time, it may be possible to more accurately predict its type [Ma and Grimson, 2005] 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.

## References

- [pets, 2001] (2001). Performance evaluation for tracking and surveillance (PETS) 2001 dataset.
- [Ablavsky and Sclaroff, 2011] Ablavsky, V. and Sclaroff, S. (2011). Layered graphical models for tracking partially-occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- [Ablavsky et al., 2008] Ablavsky, V., Thangali, A., and Sclaroff, S. (2008). Layered graphical models for tracking partially-occluded objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Andrieu et al., 2003] Andrieu, C., Freitas, N. D., Doucet, A., and Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine Learning*, 50.
- [Andriluka et al., 2008] Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Andriluka et al., 2009] Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Arie-Nachimson and Basri, 2009] Arie-Nachimson, M. and Basri, R. (2009). Constructing implicit 3D shape models for pose estimation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Atev and Papanikolopoulos, 2008] Atev, S. and Papanikolopoulos, N. (2008). Multi-view 3D vehicle tracking with a constrained filter. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [Betke et al., 2007] Betke, M., Hirsh, D., Bagchi, A., Hristov, N., Makris, N., and Kunz, T. (2007). Tracking large variable numbers of objects in clutter. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Birchfield, 2007] Birchfield, S. T. (2007). KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker. <http://www.ces.clemson.edu/~stb/klt/>
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [Bradski and Kaebler, 2008] Bradski, G. and Kaebler, A. (2008). *Learning OpenCV*. O'Reilly Media.

- [Breitenstein et al., 2011] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Gool, L. V. (2011). Online multi-person tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (to appear).
- [Breitenstein et al., 2008] Breitenstein, M. D., Sommerlade, E., Leibe, B., Gool, L. V., and Reid, I. (2008). Probabilistic parameter selection for learning scene structure from video. In *Proceedings of British Machine Vision Conference (BMVC)*.
- [Castañón, 1990] Castañón, D. (1990). Efficient algorithms for finding the k best paths through a trellis. *IEEE Transactions on Aerospace and Electronic Systems*, 26(2).
- [Comaniciu et al., 2000] Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Dahlkamp et al., 2007] Dahlkamp, H., Nagel, H.-H., Ottlik, A., and Reuter, P. (2007). A framework for model-based tracking experiments in image sequences. *International Journal of Computer Vision (IJCV)*.
- [Demirdjjan et al., 2002] Demirdjjan, D., Tollmar, K., Koile, K., Checka, N., and Darrell, T. (2002). Activity maps for location-aware computing. In *Workshop on Applications of Computer Vision*.
- [Doucet et al., 2001] Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential monte carlo methods in practice*. Springer.
- [Du and Piater, 2008] Du, W. and Piater, J. (2008). A probabilistic approach to integrating multiple cues in visual tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience.
- [Dunne and Matuszewski, 2011] Dunne, P. and Matuszewski, B. (2011). Choice of similarity measure, likelihood function and parameters for histogram based particle filter tracking in cctv grey scale video. *Image and Vision Computing*, 29.
- [Elgammal and Davis, 2001] Elgammal, A. M. and Davis, L. S. (2001). Probabilistic framework for segmenting people under occlusion. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9).

- [Fleischman et al., 2006] Fleischman, M., Decamp, P., and Roy, D. (2006). Mining temporal patterns of movement for video content classification. In *ACM workshop on multimedia information retrieval*.
- [Fleuret et al., 2008] Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30.
- [Ge and Collins, 2009] Ge, W. and Collins, R. T. (2009). Marked point processes for crowd counting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Gutchess et al., 2007] Gutchess, D., Ablavsky, V., Thangali, A., Sclaroff, S., and Snorrason, M. (2007). Video surveillance of pedestrians and vehicles. In *SPIE Conf. on Tracking, Pointing, and Laser Systems Technologies XXI*.
- [Han and Davis, 2009] Han, B. and Davis, L. S. (2009). Probabilistic fusion-based parameter estimation for visual tracking. *Computer Vision and Image Understanding (CVIU)*, 113.
- [Haritaoglu et al., 2000] Haritaoglu, I., Harwood, D., and Davis, L. (2000). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):809–830.
- [Hoiem et al., 2007] Hoiem, D., Stein, A. N., Efros, A. A., and Hebert, M. (2007). Recovering occlusion boundaries from a single image. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Irani and Anandan, 1998] Irani, M. and Anandan, P. (1998). A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- [Isard and MacCormick, 2001] Isard, M. and MacCormick, J. (2001). Bramble: A bayesian multiple-blob tracker. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Ivanov and Bobick, 2000] Ivanov, Y. A. and Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22.
- [Jepson et al., 2002] Jepson, A. D., Fleet, D. J., and Black, M. J. (2002). A layered motion representation with occlusion and compact spatial support. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Jojic and Frey, 2001] Jojic, N. and Frey, B. J. (2001). Learning flexible sprites in video layers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Kang et al., 2003] Kang, J., Cohen, I., and Medioni, G. (2003). Continuous tracking within and across camera streams. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Khan and Shah, 2006] Khan, S. M. and Shah, M. (2006). A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Khan and Shah, 2009] Khan, S. M. and Shah, M. (2009). Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- [Khan et al., 2004] Khan, Z., Balch, T., and Dellaert, F. (2004). An MCMC-based particle filter for tracking multiple interacting targets. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Khan et al., 2005] Khan, Z., Balch, T., and Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(11).
- [Kschischang et al., 2001] Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47.
- [Kumar et al., 2008] Kumar, M. P., Torr, P., and Zisserman, A. (2008). Learning layered motion segmentations of video. *International Journal of Computer Vision (IJCV)*, 76:301–319.
- [Leibe et al., 2007] Leibe, B., Leonardis, A., and Schiele, B. (2007). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision (IJCV)*.
- [Leibe et al., 2008] Leibe, B., Schindler, K., Cornelis, N., and Gool, L. V. (2008). Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30.
- [Leichter et al., 2006] Leichter, I., Lindenbaum, M., and Rivlin, E. (2006). A general framework for combining visual trackers—the “black boxes” approach. *International Journal of Computer Vision (IJCV)*, 67(3).
- [Leotta and Mundy, 2011] Leotta, M. and Mundy, J. (2011). Vehicle surveillance with a generic, adaptive, 3-D vehicle model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (to appear).
- [Leotta and Mundy, 2009] Leotta, M. J. and Mundy, J. L. (2009). Predicting high resolution image edges with a generic, adaptive, 3-d vehicle model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Leykin and Hammoud, 2006] Leykin, A. and Hammoud, R. (2006). Robust multi-pedestrian tracking in thermal-visible surveillance videos. In *CVPR Workshop on Object Tracking Beyond the Visible Spectrum*.
- [Li et al., 2009] Li, Y., Gu, L., and Kanade, T. (2009). A robust shape model for multi-view car alignment. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Lv et al., 2006] Lv, F., Zhao, T., and Nevatia, R. (2006). Camera calibration from video of a walking human. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28.
- [Ma and Grimson, 2005] Ma, X. and Grimson, W. E. L. (2005). Edge-based rich representation for vehicle classification. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Ma et al., 2009] Ma, Y., Yu, Q., and Cohen, I. (2009). Target tracking with incomplete detection. *Computer Vision and Image Understanding*, 113.
- [MacKay, 2003] MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
- [Mittal and Davis, 2003] Mittal, A. and Davis, L. S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision (IJCV)*.
- [Moore et al., 1999] Moore, D., Essa, I., and Hayes, M. (1999). Exploiting human actions and object context for recognition tasks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Nejhum et al., 2010] Nejhum, S. S., Ho, J., and Yang, M.-H. (2010). Online visual tracking with histograms and articulating blocks. *Computer Vision and Image Understanding*, 114.
- [Oliver et al., 2000] Oliver, N. M., Rosario, B., and Pentland, A. P. (2000). A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8).
- [Ottlik and Nagel, 2007] Ottlik, A. and Nagel, H.-H. (2007). Initialization of model-based vehicle tracking in video sequences of inner-city intersections. *International Journal of Computer Vision (IJCV)*.
- [Papadourakis and Argyros, 2010] Papadourakis, V. and Argyros, A. (2010). Multiple objects tracking in the presence of long-term occlusions. *Computer Vision and Image Understanding*, 114.
- [Pece, 2006] Pece, A. (2006). Contour tracking based on marginalized likelihood ratios. *Image and Vision Computing (IVC)*, 24:301–317.

- [Peursum et al., 2005] Peursum, P., West, G., and Venkatesh, S. (2005). Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Pollard and Mundy, 2007] Pollard, T. and Mundy, J. (2007). Change detection in a 3-d world. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Porikli et al., 2006] Porikli, F., Tuzel, O., and Meer, P. (2006). Covariance tracking using model update based on Lie algebra. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Rasmussen and Hager, 2001] Rasmussen, C. and Hager, G. D. (2001). Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23:560–576.
- [Reid and Connor, 2010] Reid, I. and Connor, K. (2010). Multiview segmentation and tracking of dynamic occluding layers. *Image and Vision Computing (IVC)*, 28.
- [Renno et al., 2007] Renno, J., Greenhill, D., Orwell, J., and Jones, G. (2007). Occlusion analysis: Learning and utilising depth maps in object tracking. *Image and Vision Computing*.
- [Renno et al., 2002] Renno, J., Orwell, J., and Jones, G. (2002). Learning surveillance tracking models for the self-calibrated ground plane. In *Proceedings of British Machine Vision Conference (BMVC)*.
- [Ryoo and Aggarwal, 2008] Ryoo, M. S. and Aggarwal, J. K. (2008). Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Ryoo et al., 2010] Ryoo, M. S., Lee, J. T., and Aggarwal, J. K. (2010). Video scene analysis of interactions between humans and vehicles using event context. In *CIVR*.
- [Seitz and Dyer, 1997] Seitz, S. M. and Dyer, C. R. (1997). Photorealistic scene reconstruction by voxel coloring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Senior et al., 2006] Senior, A., Hampapur, A., Tan, Y.-L., Brown, L., Pankanti, S., and Bolle, R. (2006). Appearance models for occlusion handling. *Image and Vision Computing (IVC)*, 24:1233–1243.
- [Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Sidenbladh and Wirkander, 2003] Sidenbladh, H. and Wirkander, S.-L. (2003). Tracking random sets of vehicles in terrain. In *CVPR Workshop on Multi-Object Tracking*.
- [Siebel and Maybank, 2001] Siebel, N. and Maybank, S. (2001). Real-time tracking of pedestrians and vehicles. In *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*.
- [Sigal, 2008] Sigal, L. (2008). *Continuous-state Graphical Models for Object Localization, Pose Estimation and Tracking*. PhD thesis, Brown University.
- [Sigal and Black, 2006] Sigal, L. and Black, M. (2006). Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Smith et al., 2008] Smith, K., Ba, S. O., Odobez, J.-M., and Gatica-Perez, D. (2008). Tracking the visual focus of attention for a varying number of wandering people. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30.
- [Smith et al., 2005a] Smith, K., Gatica-Perez, D., and Odobez, J.-M. (2005a). Using particles to track varying numbers of interacting people. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Smith et al., 2005b] Smith, K., Gatica-Perez, D., Odobez, J.-M., and Ba, S. (2005b). Evaluating multi-object tracking. In *CVPR EEMVCV workshop*.
- [Takala and Pietikainen, 2007] Takala, V. and Pietikainen, M. (2007). Multi-object tracking using color, texture and motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Tao et al., 2002] Tao, H., Sawhney, H., and Kumar, R. (2002). Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24:75–89.
- [Titsias, 2005] Titsias, M. (2005). *Unsupervised learning of multiple objects in images*. PhD thesis, School of Informatics, University of Edinburgh.
- [Titsias and Williams, 2006] Titsias, M. K. and Williams, C. (2006). Sequential learning of multiple objects in video. In *Sicily workshop on object recognition*.
- [Vacchetti et al., 2004] Vacchetti, L., Lepetit, V., and Fua, P. (2004). Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- [van Lieshout, 2008] van Lieshout, M. (2008). Depth map calculation for a variable number of moving objects using markov sequential object processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(7).

- [Vedula et al., 1998] Vedula, S., Rander, P., Saito, H., and Kanade, T. (1998). Modeling, combining, and rendering dynamic real-world events from image sequences. In *Fourth International Conference on Virtual Systems and Multimedia*.
- [Venkataraman et al., 2008] Venkataraman, V., Fan, X., and Fan, G. (2008). Integrated target tracking and recognition using joint appearance-motion generative models. In *CVPR workshop on Object Tracking and Classification Beyond the Visible Spectrum*.
- [Vezzani et al., 2010] Vezzani, R., Grana, C., and Cucchiara, R. (2010). Probabilistic people tracking with appearance models and occlusion classification: The AD-HOC system. *Pattern Recognition Letters*, (to appear).
- [Wang and Adelson, 1994] Wang, J. and Adelson, E. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing*, Vo. 3 No. 5:625–638.
- [White and Shah, 2007] White, B. and Shah, M. (2007). Automatically tuning background subtraction parameters using particle swarm optimization. In *IEEE International Conference on Multimedia and Expo*.
- [Winn and Blake, 2004] Winn, J. and Blake, A. (2004). Generative affine localisation and tracking. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.
- [Winn and Shotton, 2006] Winn, J. and Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Wolf et al., 1989] Wolf, J. K., Viterbi, A. M., and Dixon, G. S. (1989). Finding the best set of k paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2).
- [Wu and Nevatia, 2007] Wu, B. and Nevatia, R. (2007). Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Wu and Nevatia, 2009] Wu, B. and Nevatia, R. (2009). Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *International Journal of Computer Vision (IJCV)*, 82.
- [Xing et al., 2009] Xing, J., Ai, H., and Lao, S. (2009). Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Xu and Ellis, 2002] Xu, M. and Ellis, T. (2002). Partial observation vs. blind tracking through occlusion. In *Proceedings of British Machine Vision Conference (BMVC)*.
- [Yin and Collins, 2007] Yin, Z. and Collins, R. (2007). On-the-fly object modeling while tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Yu and Medioni, 2009] Yu, Q. and Medioni, G. (2009). Multiple-target tracking by spatiotemporal and monte carlo markov chain data association. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31.
- [Yu et al., 2008] Yu, T., Wu, Y., Krahnstoever, N. O., and Tu, P. H. (2008). Distributed data association and filtering for multiple target tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Zhou and Tao, 2003] Zhou, Y. and Tao, H. (2003). A background layer model for object tracking through occlusion. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Zhu et al., 2008] Zhu, L., Zhou, J., and Song, J. (2008). Tracking multiple objects through occlusion with online sampling and position estimation. *Pattern Recognition*, 41:2447–2460.

# Curriculum Vitae

Vitaly Ablavsky

## Personal Info

Citizenship: U.S.

Languages: English, Russian (native)

## Contact Info

Boston University  
Department of Computer Science  
111 Cummington St.  
Boston, MA 02215

Phone (lab.) 1 617 353 9777

Phone (dept. office) 1 617 353 8919

Fax 1 617 353 6457

e-mail: [ablavsky@cs.bu.edu](mailto:ablavsky@cs.bu.edu)

web: <http://cs-people.bu.edu/ablavsky>

## Education

- Boston University, Computer Science Dept.: Ph.D. (2011)
- University of Massachusetts Amherst, Computer Science Dept.: M.S. (1996)
- Brandeis University: B.A. (1992), major in Mathematics

## Professional Appointments

2005 - present: Research Assistant and Teaching Fellow, CS Dept., Boston University  
2002 - 2005: Principal Research Engineer, Charles River Analytics, Inc., Cambridge, MA  
1999 - 2002: Senior Software Engineer, Charles River Analytics, Inc., Cambridge, MA  
1998 - 1999: Software Engineer, Cognex Corporation, Natick, MA  
1996 - 1998: Senior Software Engineer, Amerinex Applied Imaging, Amherst, MA

## Professional Activities

- reviewer for AVSS, CIVR, CVPR, ECCV, ICPR, ICCV, IJPRAI, NIPS, and VSSN
- captain of IVC student-volunteers for AVSS 2010
- co-organizer of IVC weekly research colloquiums 2008-2009

## Publications

### Publications in Refereed Journals:

1. Vitaly Ablavsky and Stan Sclaroff, “Layered Graphical Models for Tracking Partially-Occluded Objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, Vol. (to appear), 2011.
2. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky, and Stan Sclaroff, “Learning a Family of Detectors via Multiplicative Kernels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, Vol. (to appear), 2011.

### Publications in Refereed Proceedings:

1. Vitaly Ablavsky, Ashwin Thangali, and Stan Sclaroff, “Layered graphical models for tracking partially-occluded objects,” In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
2. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky, and Stan Sclaroff, “Multiplicative Kernels: Object Detection, Segmentation and Pose Estimation,” In Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2008.
3. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky, and Stan Sclaroff, *Parameter Sensitive Detectors*, In Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2007.
4. Daniel Gutchess, Vitaly Ablavsky, Ashwin Thangali, Stan Sclaroff and Magnús Snorrason, “Video Surveillance of Pedestrians and Vehicles,” In Proc. SPIE Conf. on Tracking, Pointing, and Laser Systems Technologies XXI, Vol. 6569, 2007.
5. Camille Monnier, Vitaly Ablavsky, Stephen Holden, and Magnús Snorrason, “Sequential Correction of Perspective Warp in Camera-based Documents,” In Proc. IEEE International Conference on Document Analysis and Recognition (ICDAR), Seoul, Korea, August 2005.
6. Vitaly Ablavsky, “Background Models for Tracking Objects in Water,” In Proc. IEEE International Conference on Image Processing (ICIP), Barcelona, Spain, September 2003.
7. Vitaly Ablavsky and Mark R. Stevens, “Automatic Feature Selection with Applications to Script Identification of Degraded Documents,” In Proc. IEEE International Conference on Document Analysis and Recognition (ICDAR), Edinburgh, UK, pp. 750-754, August 2003.
8. Vitaly Ablavsky, Daniel Stouch, and Magnús Snorrason, “Search Path Optimization for UAVs using Stochastic Sampling with Abstract Pattern Descriptors,” In Proc. AIAA Guidance Navigation and Control Conference, Austin, TX, August 2003.

9. Thomas Goodsell, Magnús Snorrason, Mark R. Stevens, Dustin Cartwright, Brian Stube, and Vitaly Ablavsky, "Sign Detection for Autonomous Navigation," In Proc. SPIE Unmanned Ground Vehicle Technology V, 2003.
10. Vitaly Ablavsky, Magnús Snorrason, and Colin J. Taylor, "RAVE: Real-time Autonomous Video Enhancement system," In Proc. IEEE International Conference on Image Processing (ICIP), Rochester, NY, September 2002.
11. Vitaly Ablavsky, Magnús Snorrason, and Stephen Holden, "Efficient Pursuit of a Moving Target via Spatial Constraint Exploitation," In Proc. AIAA Guidance, Navigation, and Control Conference, Montreal, CA, August 2001.
12. Thomas Goodsell, Magnús Snorrason, Harald Ruda, and Vitaly Ablavsky, "Navigability Evaluation and Visualization for Mars Rover Operations," In Proc. the International Conference on Visualization, Imaging and Image Processing, Marbella, Spain, July 2001.
13. Mark R. Stevens, Magnús Snorrason, Vitaly Ablavsky, and Sengvieng Amphay, "ATA Algorithm Suite for Co-Boresighted PMMW and LADAR Imagery," In Proc. SPIE, Volume 4373, AeroSense, Orlando, FL, April 2001.
14. Thomas Goodsell, Magnús Snorrason, Harald Ruda, and Vitaly Ablavsky, "Automated Obstacle Mapping and Navigability Analysis for Rover Mission Planning," In Proc. SPIE, Volume 4364, AeroSense, Orlando, FL, April 2001.
15. Thomas Goodsell, Magnús Snorrason, Harald Ruda, and Vitaly Ablavsky, "Rover Obstacle Visualizer and Navigability Evaluator," In Proc. SPIE, Volume. 4195, Photonics East, Boston, MA, November 2000.
16. Vitaly Ablavsky and Magnús Snorrason, "Optimal Search for a Moving Target: a Geometric Approach," In Proc. AIAA Guidance, Navigation, and Control Conference, Denver, CO, August 2000.

#### **Miscellaneous Publications and Reports:**

1. Camille Monnier, Vitaly Ablavsky, Stephen Holden, and Magnús Snorrason, "A Document Image Enhancement Module: Perspective Warp Correction," In Proc. Symposium on Document Image Understanding Technology (SDIUT), College Park, MD, November 2005.
2. Vitaly Ablavsky, Magnús Snorrason, and Mark R. Stevens, "OCR Accuracy Prediction as a Script Identification Problem," In Proc. Symposium on Document Image Understanding Technology (SDIUT), Greenbelt, MD, April 2003.
3. Vitaly Ablavsky, Mark R. Stevens and Joshua Pollak, "Data-Structure-Independent Algorithms for Image Processing," C/C++ Users Journal, pp. 24-31, November 2003.

4. Vitaly Ablavsky, "Applying BGL to Computational Geometry," *C/C++ Users Journal*, pp. 6-12, August 2002.
5. Vitaly Ablavsky, Dustin Cartwright, Magnús Snorrason, Mark R. Stevens, and Joshua Pollak, "ISIS: Intelligent Surveillance and Intrusion Detection for Ships," Final Technical Report R02131, Charles River Analytics Inc., Cambridge, MA, October 2002.
6. Magnús Snorrason, Vitaly Ablavsky, and Colin J. Taylor, "Passive Millimeter-Wave and Laser-Radar Autonomous Target Acquisition," Final Technical Report R99079, Charles River Analytics Inc., Cambridge, MA, January 2000.

### **Teaching Experience**

Teaching Fellow for:

- CS212 (Spring 2006, 2007) undergraduate course in C# programming for the .NET platform
- CS440/640 (Fall 2007) undergraduate/graduate course in Artificial Intelligence
- CS542 (Spring 2009) undergraduate/graduate course in Machine Learning

### **Research Interns Supervised:**

- at IVC: Kyle Olszewski (B.A., Computer Science, Boston University)
- at Charles River Analytics, Inc.: Vakhid Masagutov (Ph.D., Mathematics, Purdue), Dustin Cartwright (A.B., Mathematics, Harvard), Marc Richards (M.S., Computer Science, Colorado State), Stephen Holden (B.Eng., Computer Engineering, Memorial University of Newfoundland)